ヒダピオ制御カー

専用ソフトウェア「ArduBlock_HCC」

取扱説明書 0.1

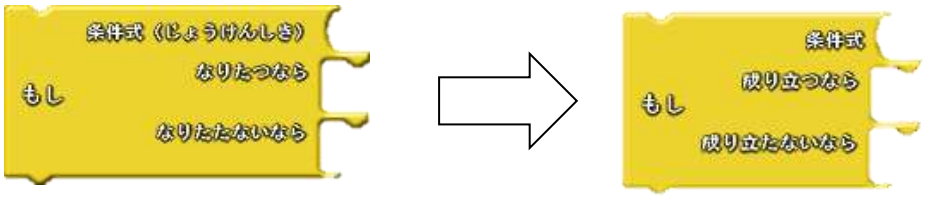
トップマン

おことわり

説明書内と製品では下記内容が異なります。

■ブロック内の表示について

製品ではひらがな表記はなくなっています。
例



■動きブロック（7-2 動き・LED の項参照）

設定できる方向が増えています。

「前進」「後退」「右折」「左折」「停止」

↓

「前進」「後退」「右前進」「右旋回」「右後退」
「左前進」「左旋回」「左後退」「停止」

■入力ブロック（7-3 入力の項参照）

「中央明るさセンサ」が「光センサ」に変更。

■ポート出力ブロックの追加

【注意】方向設定ブロック（7-2 動き・LED の項参照）との併用では動作しません。

これらの変更に関しては近日中に説明書を修正いたします。
今しばらくお待ちください。
宜しくお願い致します。

INDEX

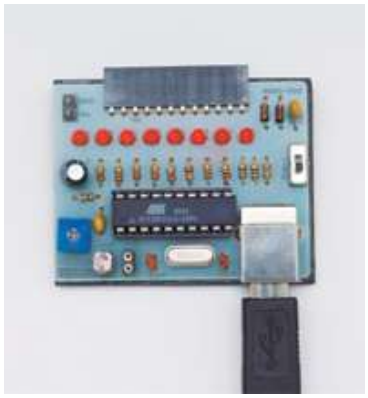
1　はじめに 4
1-1　プログラム書込みライター 4
1-2　ソフトウェアのダウンロード 5
1-3　書込みの為の初期設定 7
1-4　本体の説明 9
2　簡単なプログラムの作成 10
2-1　ソフトウェアの起動 10
2-2　ファイルの新規作成、開く ..... **エラー! ブックマークが定義されていません。**
2-3　画面の説明 11
2-4　ブロックの追加 11
2-5　ブロックの組立て 12
2-6　プログラムの書込み 13
2-7　ブロックの詳細設定 15
2-8　プログラムの保存・読込み 18
3　くり返しプログラムの作成 22
4　分岐プログラムの作成 23
4-1　リモコン入力の確認 23
5　音を鳴らすプログラム 32
6　センサー基板を使ったプログラム 33
7　上級者向けプログラム 36
8　便利な機能 53
9　各ブロックの説明 56

1 はじめに

この度は、「ヒダピオ制御カー」（以下、「制御カー」と記述）をご使用頂きありがとうございます。制御カー専用ソフトウェア「ArduBlock_HCC」（以下、「ArduBlock」と記述）は、制御カーと接続してプログラムを作成・書込みを行うためのソフトウェアです。「ソフトウェア取扱説明書」（以下、「本書」と記述）は、ソフトェアの操作方法について説明した資料です。

1-1 プログラム書込みライター

制御カーにプログラムを書き込むには書込みライター「ヒダピオ学習回路」が別途必要です。



ヒダピオ学習回路

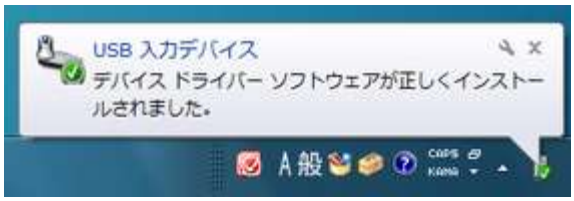
パソコンとの接続

ヒダピオ学習回路はライターモードにして使用します。

スイッチを写真の位置（上側）にします。

パソコンに接続して右から 3 番目の LED が点灯していることを確認します。

初めてパソコンに接続した際に画面の右下に以下のようなメッセージが表示されます。



※Windows7の場合

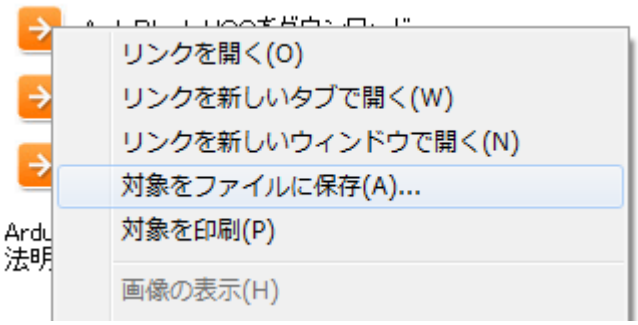
このメッセージが表示されれば認識は完了です。

（注）メッセージが表示されるまではパソコンから取り外さないでください。正常に認識されなくなる可能性があります。

また、このメッセージが表示されるのは最初の一回目だけです。正常に認識された後、2 回目以降は表示されません。

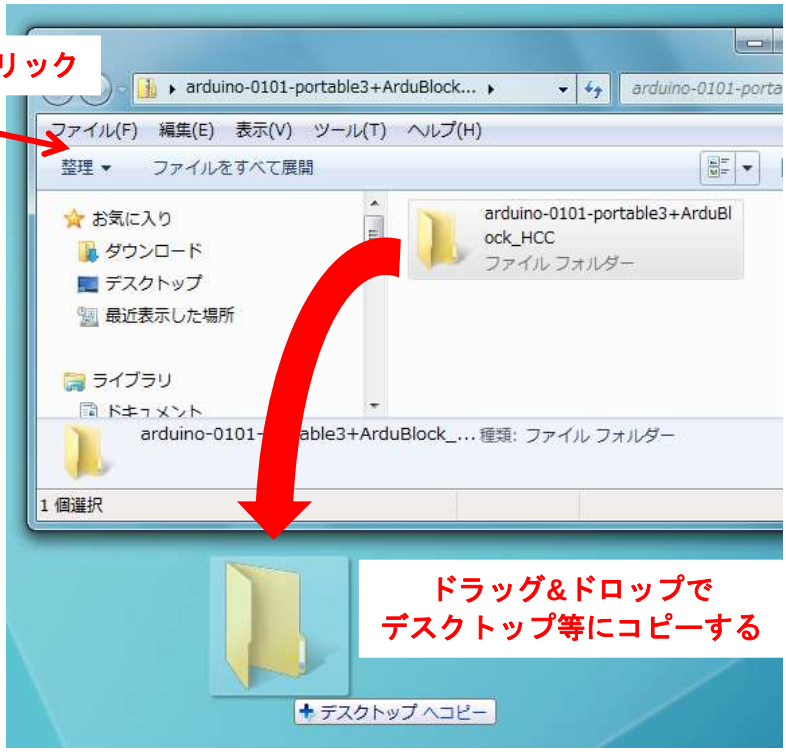
1-2 ソフトウェアのダウンロード

ホームページ中ほどにある下図の矢印アイコンを右クリックし、「対象をファイルに保存」を選択します。



ダウンロードしたファイルは ZIP 形式で圧縮されているので解凍してください。解凍後に出来たフォルダを丸ごとパソコンの好きな場所（デスクトップなど）にコピーしてください。

ダウンロードした
ZIP ファイル



コピーしたフォルダ

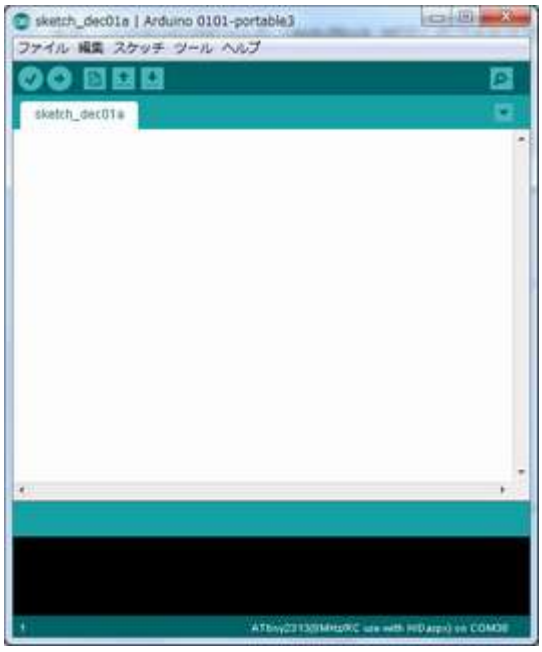
1-3 書込みの為の初期設定

以下の設定を制御カー1 台につき最初の一回だけ行います。

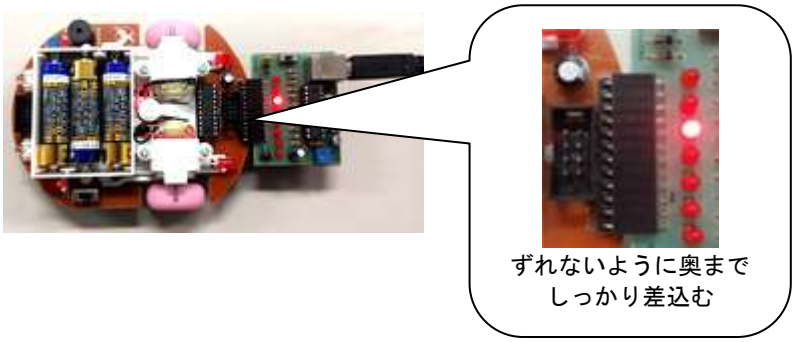
先ず、パソコンにコピーした「ArduBlock_HCC」のフォルダを開いてください。フォルダを開くと中に「arduino.exe」という実行ファイルが入っているので、ダブルクリックして実行してください。

「ArduBlock_HCC」のフォルダを開いて
「arduino.exe」をダブルクリックする。

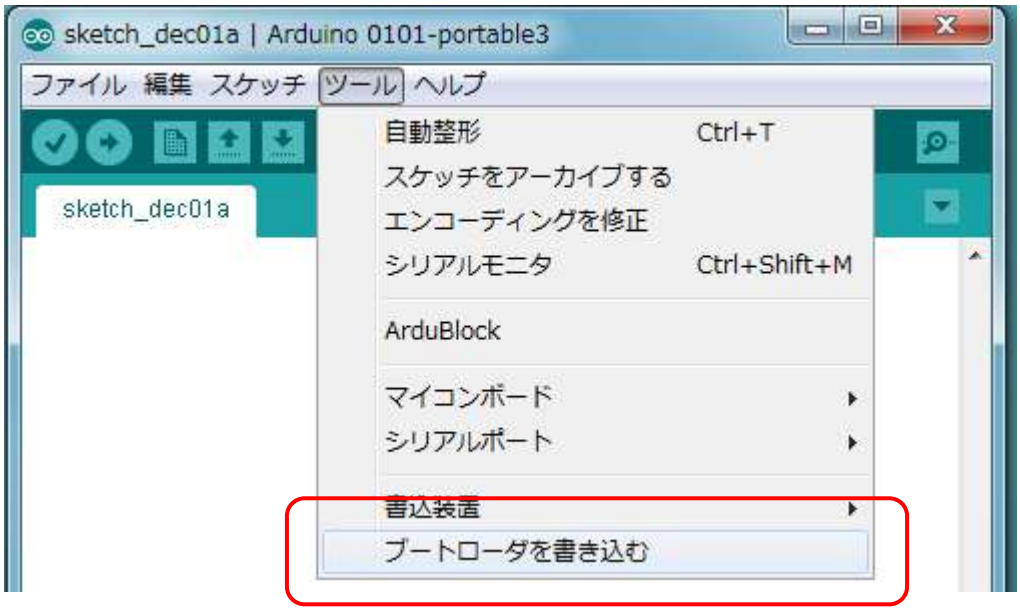
ソフトウェアが起動し、画面に下記のウインドウを表示します。



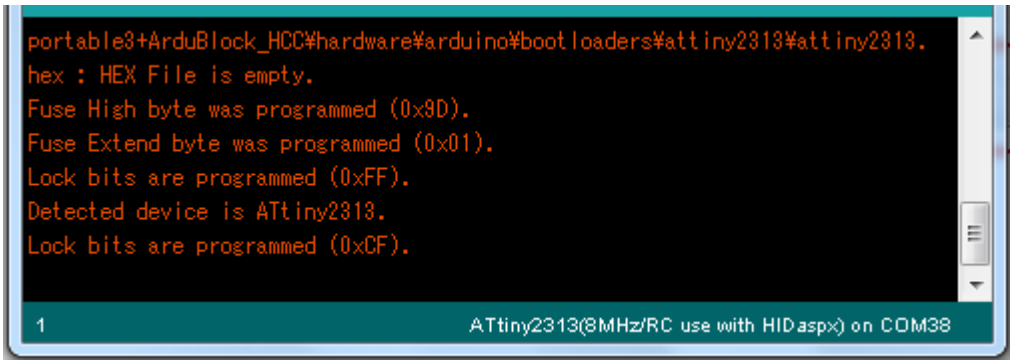
ヒダピオ学習回路（以降「書込みライター」と記述）を制御カーに接続します。



ブートローダーを書き込むをクリックします。



以下のようなメッセージが表示されれば設定は終了です。

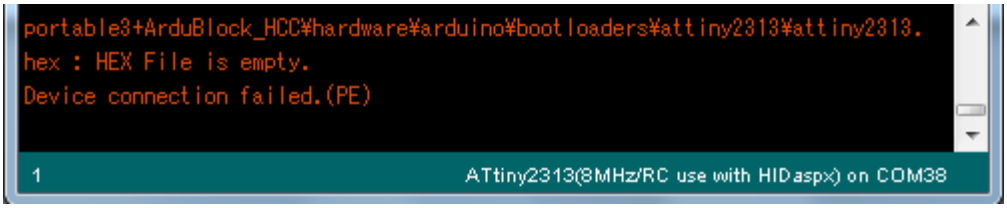


以下のようなメッセージが表示された場合は以下の内容を確認して、再度書込みを行って下さい。

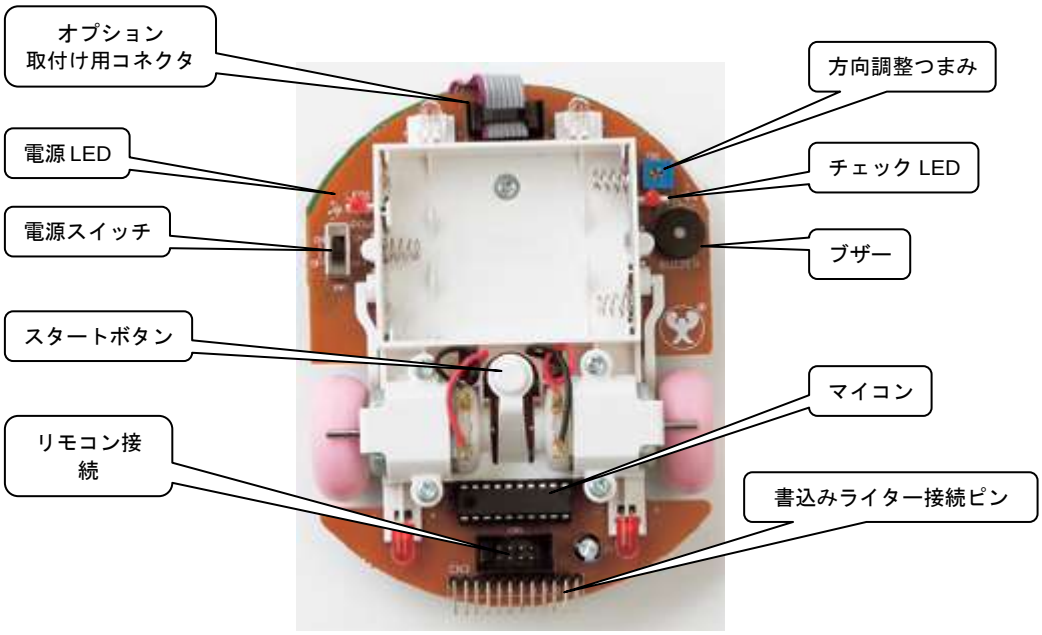
・書込みライターがライターモードになっているか確認する。

・制御カーと書込みライターを接続し直す。

・パソコンと書込みライターを接続し直す。



1-4 本体の説明



2 簡単なプログラムの作成

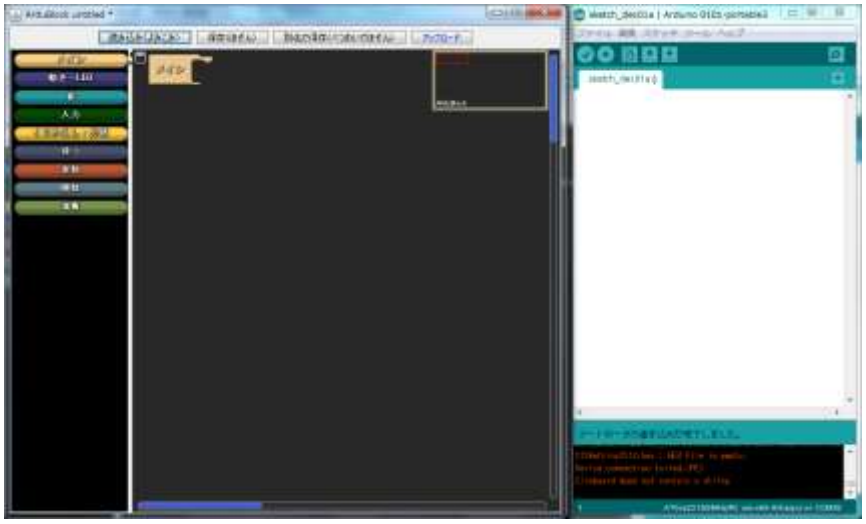
2-1 ソフトウェアの起動

パソコンにコピーした「ArduBlock_HCC」のフォルダを開いてください。フォルダを開くと中に「arduino.exe」という実行ファイルが入っているので、ダブルクリックして実行してください。

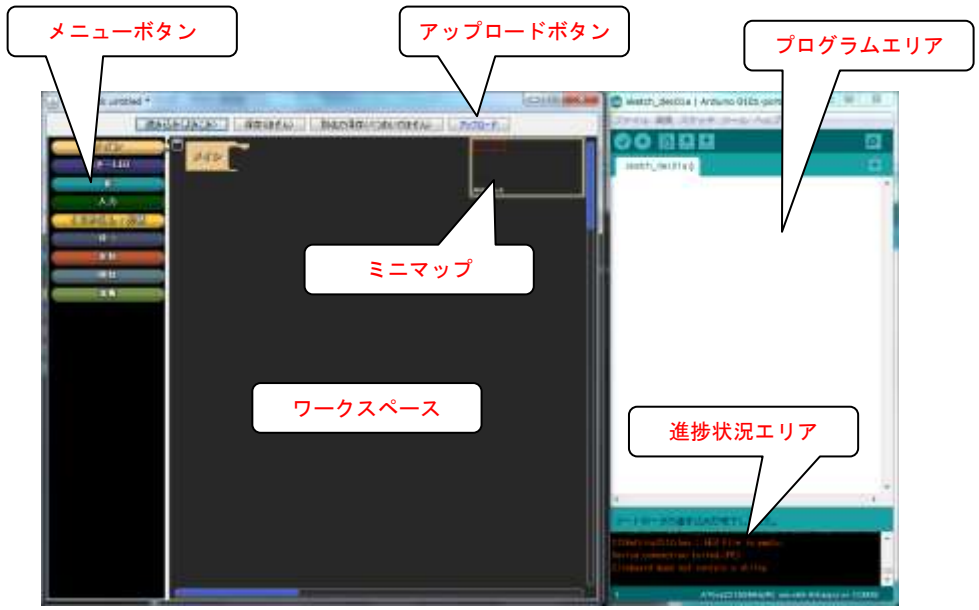
「ArduBlock_HCC」のフォルダを開いて
「arduino.exe」をダブルクリックする。

「ツール」→「ArduBlock」をクリックします。

プログラム作成用の画面が表示されます。



2-2 画面の説明



2-3 ブロックの追加

例に制御カーが前進するだけの簡単なプログラムを作成してみます。
まず、メニューボタンの中から「動き・LED」のボタン（下図参照）をクリックしてください。新たに表示されたブロックから一番上のブロックをワークスペースにドラッグ&ドロップ(※)してください。ワークスペースにに選択したブロックが追加されます。

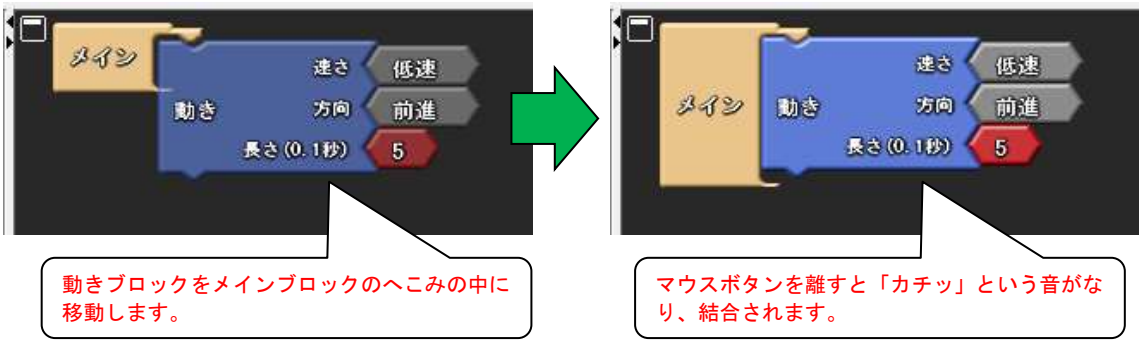
※マウスポインタがブロックに重なった状態でマウスのボタンを押し、そのままの状態でマウスを移動させ、ワークスペース内でマウスのボタンを離してください。

プログラムで使用するブロックは、このように「使用するブロックの種類をメニューボタンから選び」→「ワークスペースへドラッグ&ドロップ」という手順で追加します。

2-4 ブロックの組立て

これで「制御カーを前進させる」ブロックがプログラムに追加されましたが、このままではプログラムを書き込んでも制御カーは動きません。なぜなら、「メインブロック」の中に追加した前進のブロックが組み込まれていないためです。追加したブロックを制御カーで実行させる場合は、それぞれのブロックをメインブロックの中に接続して組み込む必要があります。

動きブロックをメインブロックに接続する場合は、動きブロックをメインブロック内に移動(ドラッグ)し、マウスボタンを離すと「カチッ」という音と共にメインブロックに接続されます。

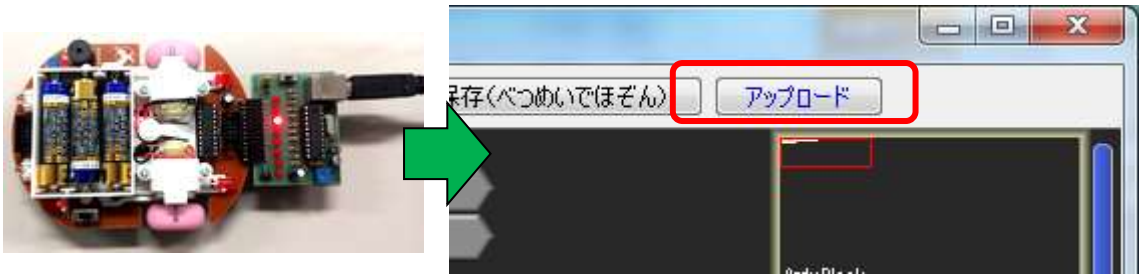


2-5 プログラムの書込み

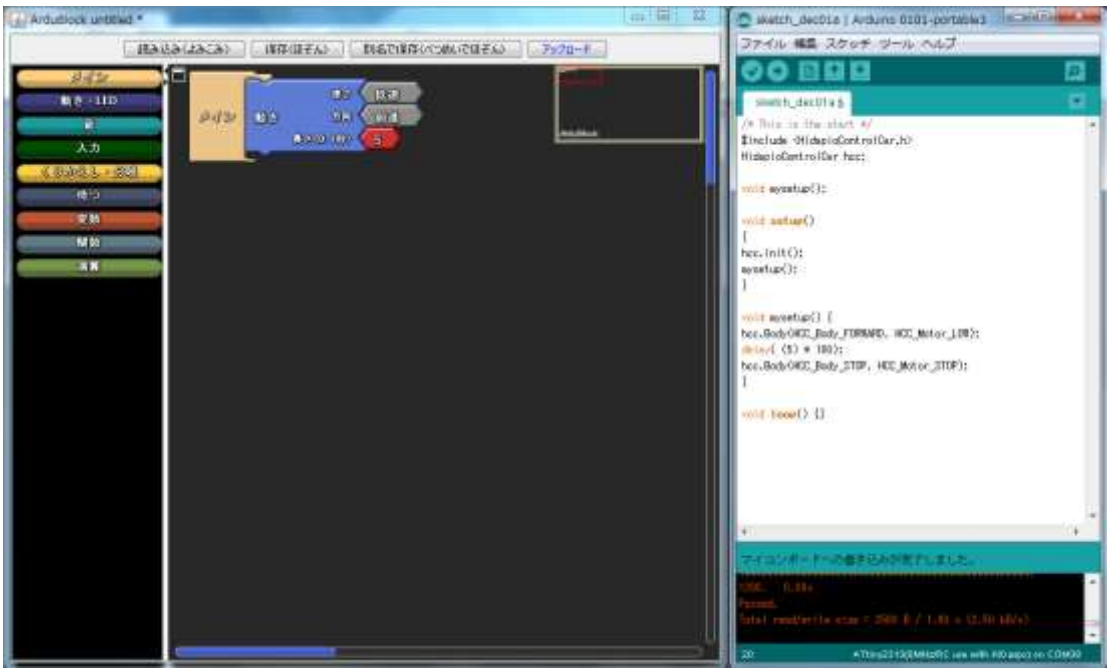
作成したプログラムを制御カーに書き込んで実行させてみましょう。

※プログラムの書き込みは、制御カーの電源ボタンを OFF にして行って下さい。

まず、制御カーに書込みライターを接続して、画面上の「アップロード」ボタンをクリックしてください。

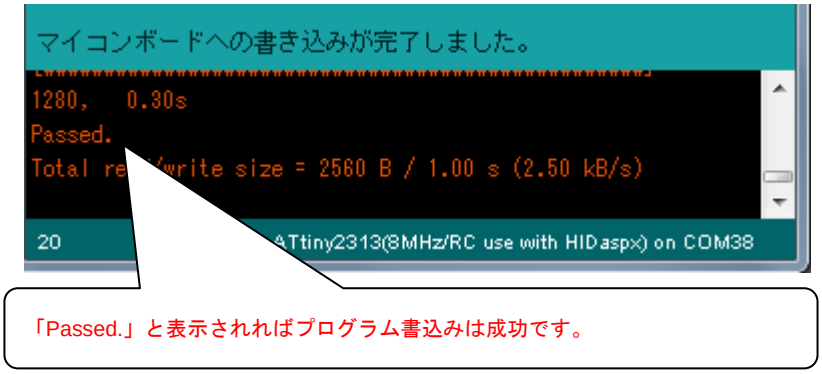


プログラムエリアにプログラムが生成され、コンパイル、プログラム書込みが行われます。

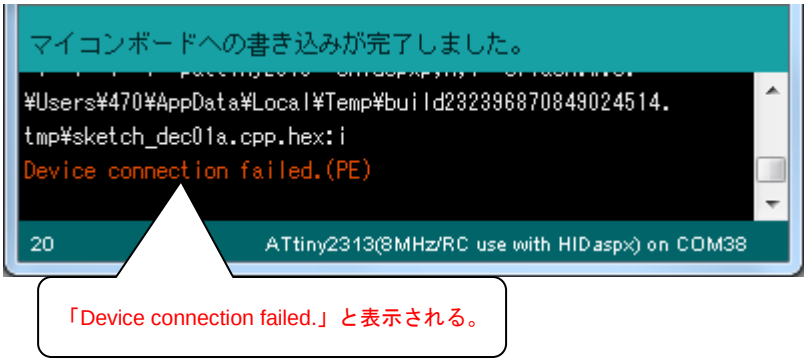


プログラムの書き込み中は、書込みライターを制御カーから取り外さないでください。

進捗状況エリアに下図のメッセージが表示されればプログラム書込みは終了です。



正常にプログラムを書き込みできなかった場合、下記のメッセージが表示されます。この場合、書込みライターを挿し直してもう一度正しい手順で書き込みを行ってください。



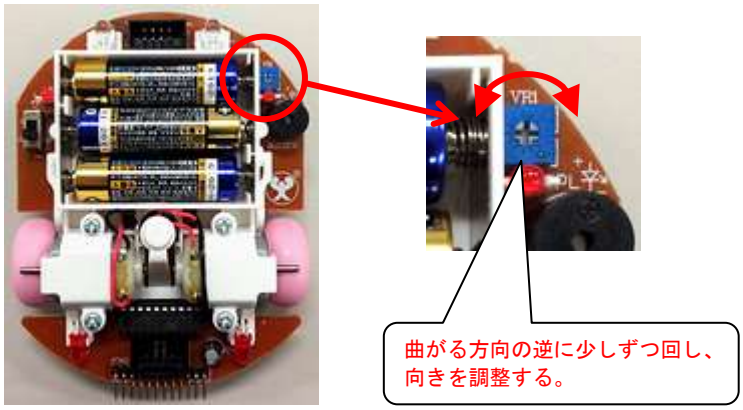
書き込んだプログラムを実行してみます。書込みライターから取り外した状態で電源スイッチを ON にすると、すぐにプログラムを開始します。制御カーが動いたときに机から落ちたりしないよう、安全な場所に置いてください。

準備ができたら電源スイッチを ON にして手を離してください。
作成したプログラムの通り制御カーが約 0.5 秒前進して止まります。
最後までプログラムを実行すると制御カーは停止するので、もう一度プログラムを開始する場合は、一度電源スイッチを OFF にして再び ON にしてください。また、途中でプログラムを止める場合は、電源スイッチを OFF にしてください。

2-6 モーターの調整

プログラムを実行したときに、右もしくは左に微妙に曲がる場合があります。これは、モーターの個体差や動かす際の地面などの影響によるものです。今回は、ロボットが真っ直ぐ走ることが理想なので、左右のモーターのスピードを調整して、真っ直ぐ走るようにします。

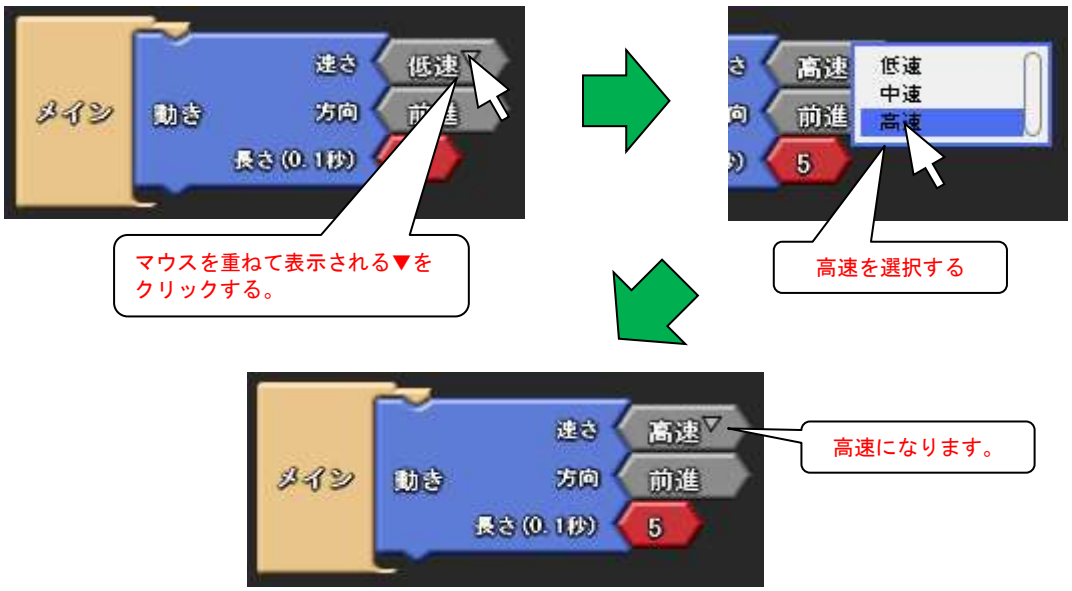
先ず、プログラムを続けて実行して、左右どちらに曲がるか確認します。その後、制御カー本体の「方向調整つまみ」を曲がるのとは逆方向にほんの少しずつ回し、真っ直ぐに進む位置に調整します。



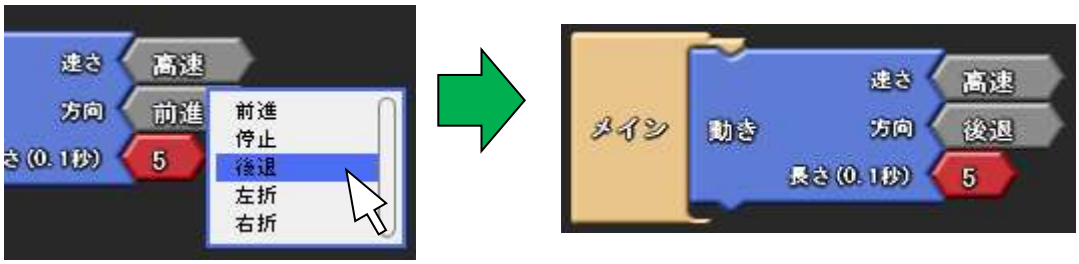
2-7 ブロックの詳細設定

各ブロックは「時間を変える」など、細かい設定ができるようになっています。先に使用した動きブロックの設定を変更してみます。

動きブロックは制御カーの進む「速さ」「方向」「長さ(時間：0.1 秒単位)」を設定できます。
速さを変更する場合は「低速」の上にマウスを移動させると▼マークが表示されます。▼マークをクリックすると速度を選択できます。ここでは「高速」を選択します。



同様に方向を「後退」にします。



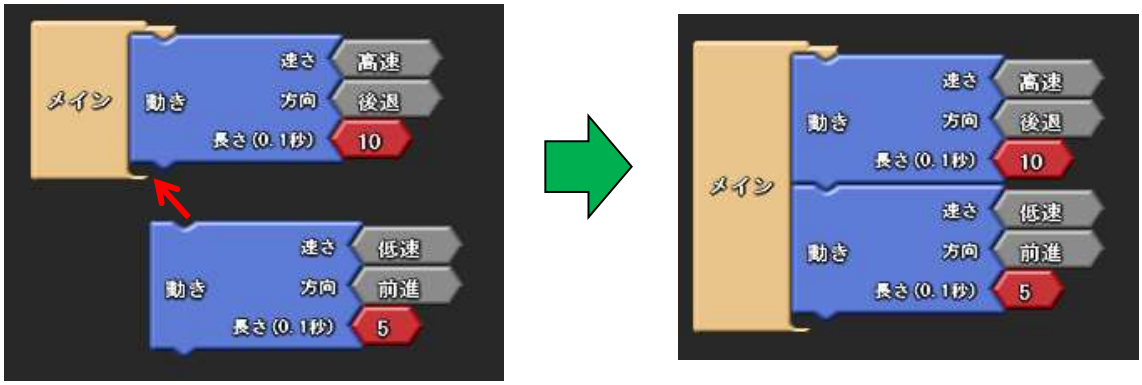
長さは数字をクリックすると入力できるようになります。
時間の単位は 0.1 秒です。ここでは 10（1 秒）にします。

設定が終われば再び制御カーにプログラムを書き込み、動きが変わった事を確認してください。

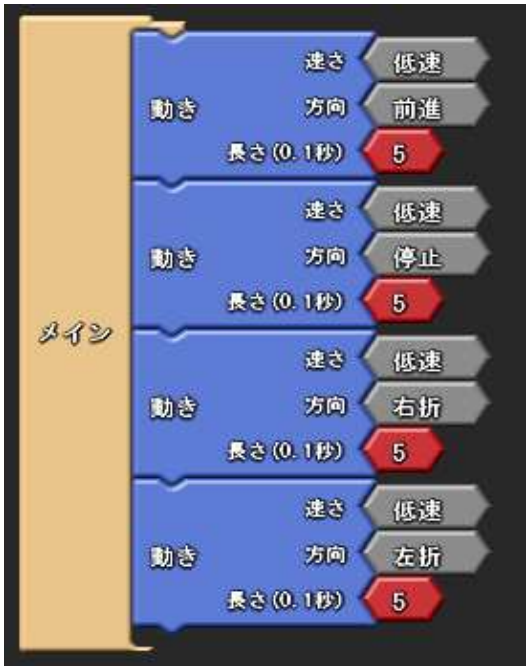
2-8 他のブロックの追加

次に複数のブロックを含んだプログラムを作成してみます。先ほどのプログラムに、「停止」「右折」「左折」を追加ましょう。

動きブロックをもう一つ追加し、先の動きブロックの下に接続します。



同様にあと二つの動きブロックを追加し、接続します。そして上から「前進」「停止」「右折」「左折」に設定します。速さは全て「低速」、長さは「5」にしましょう。

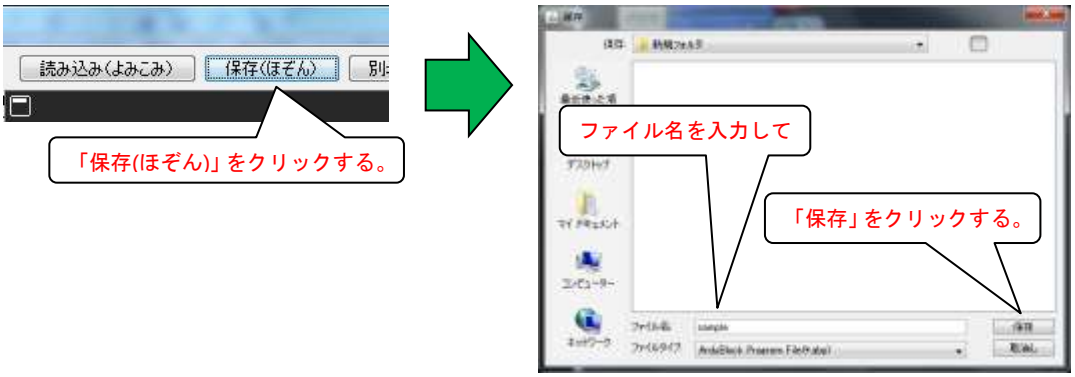


設定が終われば再び制御カーにプログラムを書き込み、動きが変わった事を確認してください。

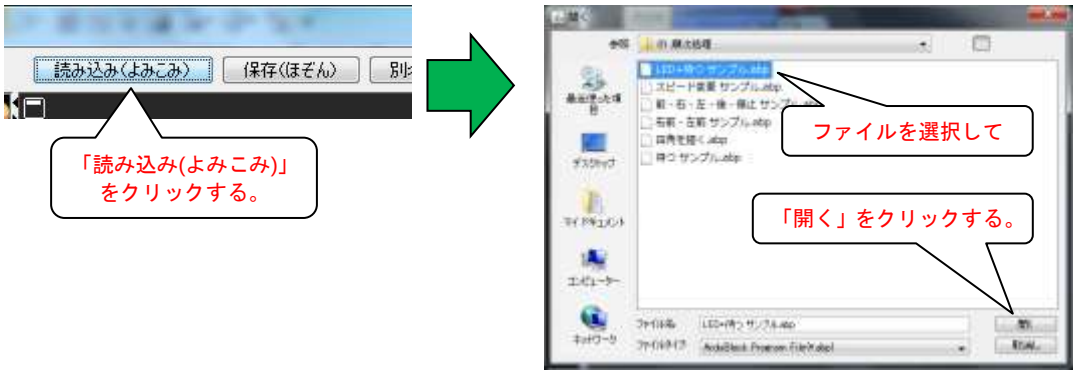
2-9 プログラムの保存・読込み

作成したプログラムは、ファイルに保存したりファイルから読み込んだりすることができます。プログラムをファイルに保存する場合は、次の手順で行います。

プログラムをファイルに保存する場合は、上部メニューより「保存(ほぞん)」をクリックしてください。クリックすると右下図のような画面が表示されるので、プログラムのファイル名を入力して「保存」をクリックしてください。

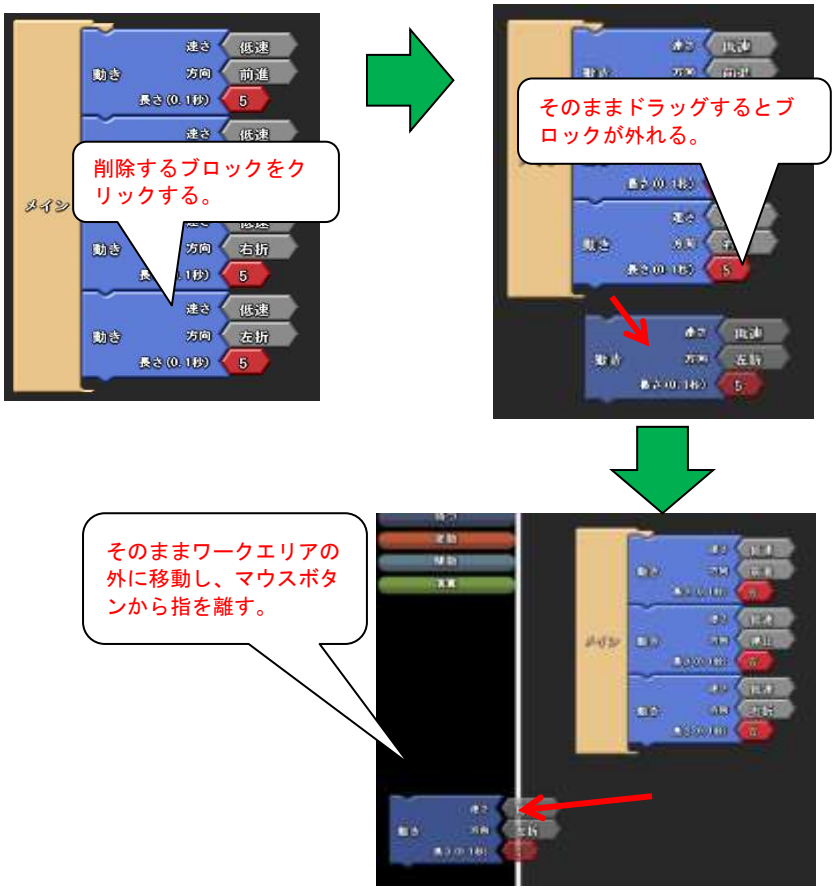


また、ファイルからプログラムを読み込む場合は、メニューの「読み込み(よみこみ)」をクリックします。クリックすると右下図のような画面を開くので、読み込むファイル名を入力して「開く」をクリックしてください。

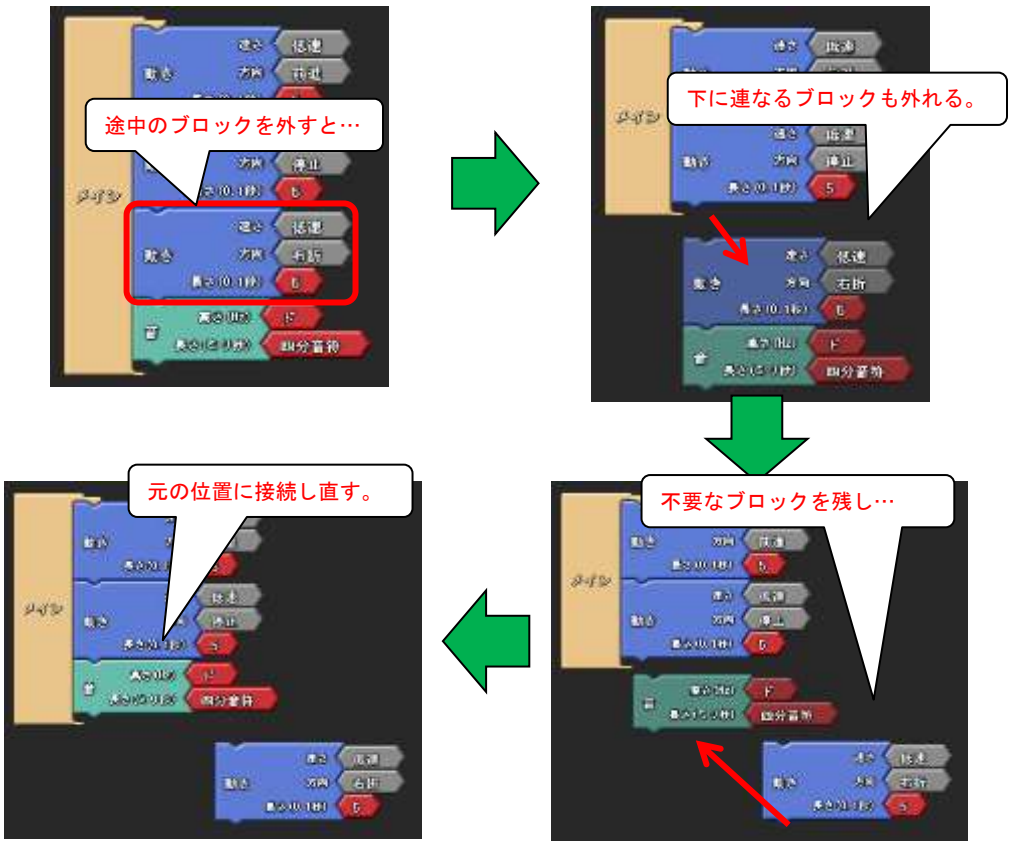


2-10 ブロックの削除

ブロックをワークエリアから削除する場合は、以下の通りにします。
削除するブロックをクリックし、そのままドラッグします。ブロックが外れたらそのままワークエリア外に移動させマウスボタンから指を離すとブロックが削除されます。



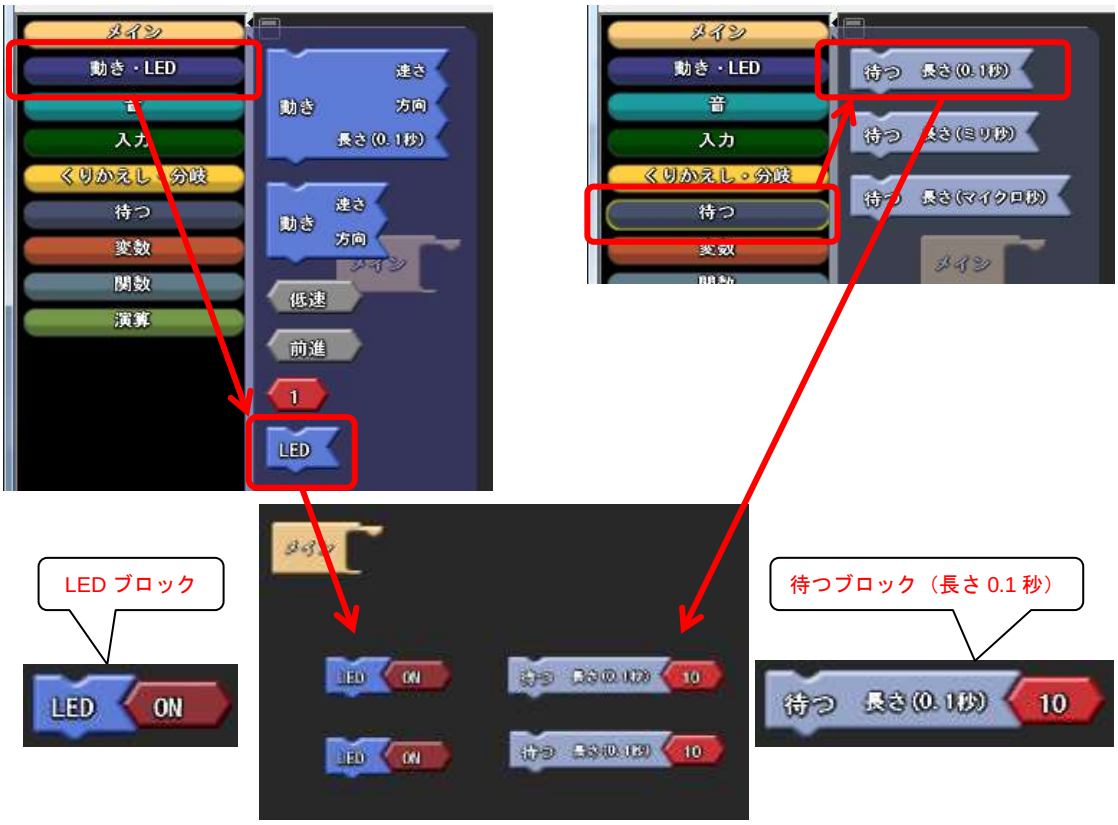
下につながっているブロックは一緒に外れます。
一旦任意の場所に移動して、目的のブロックを残し、再度メインブロック内の元の位置に接続してください。



2-11 LED ブロック

次に LED を 1 秒間隔で点滅させるプログラムを作成します。
現在作成しているプログラムからメインブロック以外のブロックを消してください。

次にメニューボタンから下図の「LED ブロック」を 2 つ、「待つ 長さ(0.1 秒)ブロック」を 2 つ追加してください。



これらのブロックを下左図のように接続し、下の「LED ブロック」を「OFF」にします。

プログラムが完成したら制御カーに書き込みます。

LED は書込み終了後に即動作します。書込みライターを取り外すまでプログラムはリセットされません。再度実行する場合は書込みライターを取り外し、スイッチを ON にしてください。

3 くり返しプログラムの作成

今まで作ったプログラムは１回だけ実行して終了していました。これを任意の回数繰り返すようにするには「くり返しブロック」を使用します。

「くり返しブロック」には２種類あります。「ずっとくりかえす」と書かれたブロックは、電源を切るまで接続されたプログラムを繰り返し続けます。「くりかえす」と「回数」が書かれたブロックはくり返し回数を決める事ができます。

「くり返しブロック」を使ったプログラムは以下のようになります。

ここまでの説明で、モーターや LED を自由に動かすプログラミングができるようになりました。しかし、まだ決められた一本道のプログラムしか作ることができません。

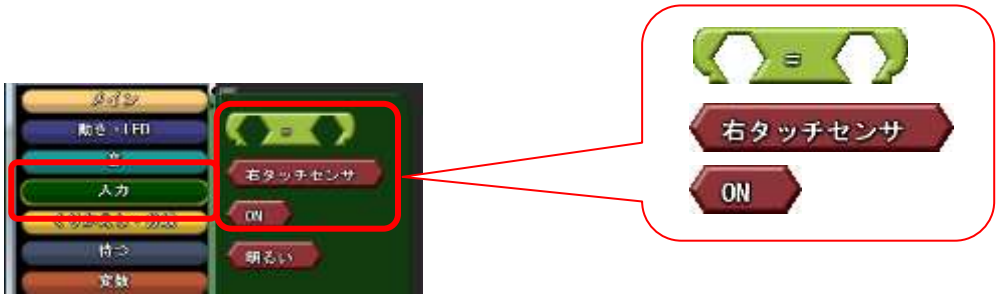
次項では条件によって制御カーの動きを変える条件分岐のプログラミングを作ります。

4 リモコンを使ったプログラムの作成

制御カーには専用のリモコンが付いています。リモコンには「FORWARD（前進）」「BACK（後退）」「RIGHT（右折）」「LEFT（左折）」「START（スタート）」のボタンが付います。（「START」ボタンは本体にも付いています。機能はリモコンと共通です。）これらを用いて「条件分岐」のプログラムを作ってゆきます。

4-1 ボタン入力の確認

本体の「START ボタン」を押せば LED が光るプログラムを作成します。先ずメニューボタンの「入力」から下図の 3 つのブロックをワークスペースに配置します。

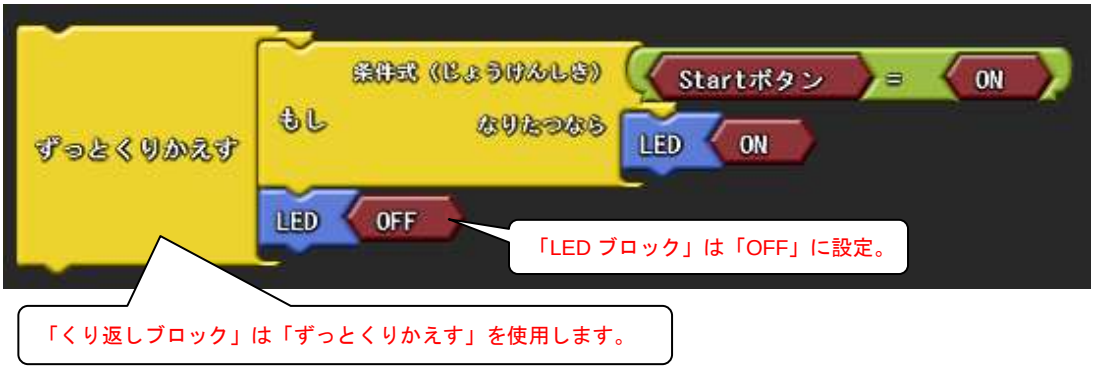


「右タッチセンサ」を「START ボタン」に変更し下図のように接続します。

次に条件を設定する「もしブロック」をワークスペースに配置します。そして先ほどの接続済みのブロック群を下図のように「もしブロック」に接続します。

更に「LED ブロック」を下図のように接続します。

そして、「くり返しブロック」ともう一つ「LED ブロック」を下図のように接続します。LED ブロックは「OFF」に設定します。



最後に「メインブロック」に接続します。

これでプログラムは完成です。制御カーにプログラムを書き込んで動作を確認します。このプログラムでは「START ボタン」を押している間だけ LED を点灯させるようになっています。制御カーの「START ボタン」を押して、動作を確認してください。

4-2 リモコンを使ったプログラム

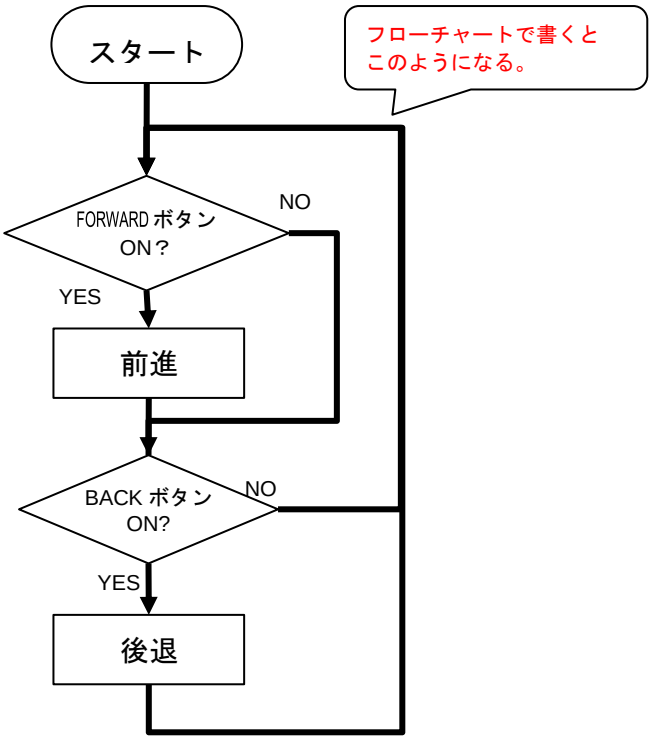
それではリモコンを使ったプログラムを作成してゆきます。ここではリモコンの方向ボタンを押した時にその方向に進むプログラムを作成します。
先ず「FORWARD（前進）ボタン」を押した時に制御カーが前進するプログラムは以下のようになります。

プログラムを書き込んでリモコンを接続し、動作を確認してください。「FORWARD ボタン」を押している間、前進します。

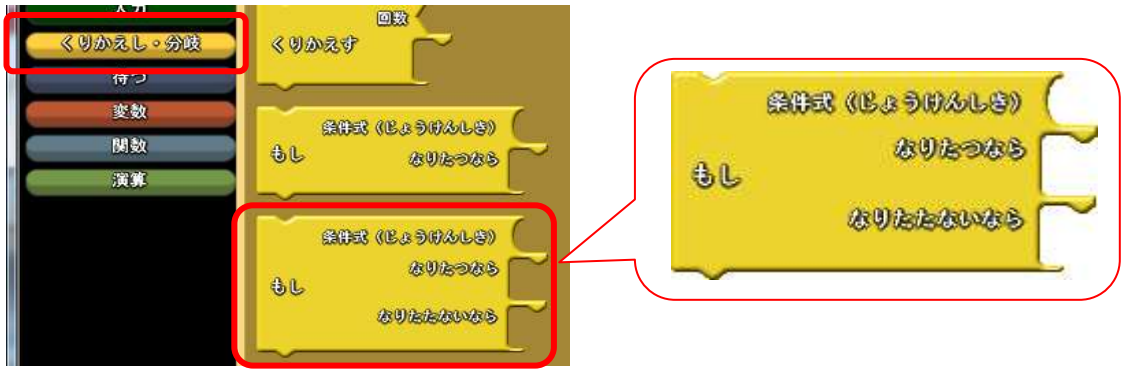
リモコンを使ったプログラムの作成

次に「BACK（後退）ボタン」を押した時の動作を追加します。「BACK ボタン」を押している間、制御カーを後退させるようにします。下図のようにブロックを追加してください。

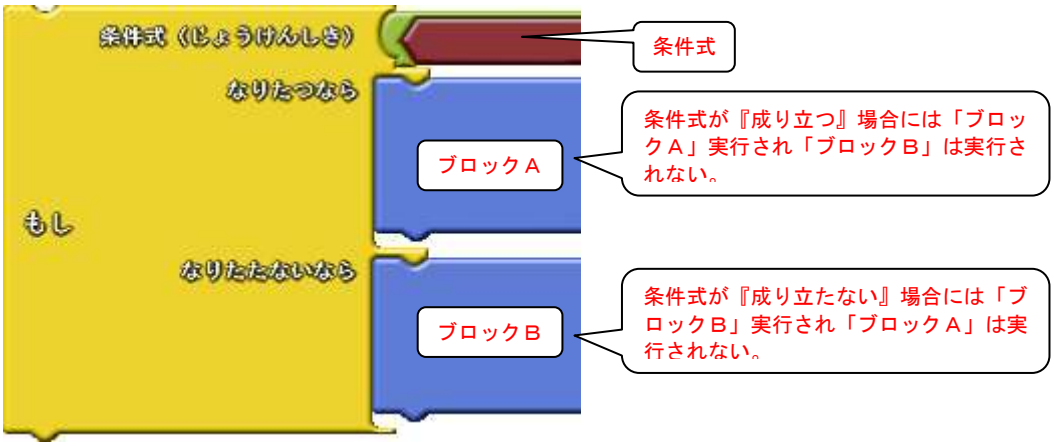
プログラムを書き込んで動作を確認してください。このプロフラムでは「くり返しブロック」の中に「もしブロック」が 2 つ続けて接続されています。プログラムが開始すると、先ず最初の「もしブロック」で「FORWARD ボタン」が押されているかをチェックします。押されていれば前進の「動きブロック」を実行し、次に進みます。押されていなければ何もせずに次に進みます。その後、次の「もしブロック」で「BACK ボタン」が押されているかをチェックし、先と同じ動作を実行します。これらは「くり返しブロック」でくくられているので、電源を切るまでくり返しつづけます。

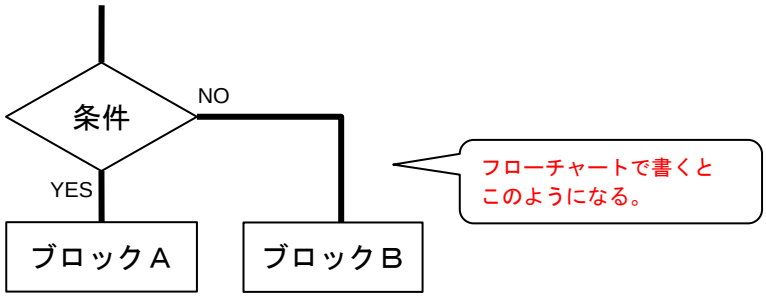


では、「FORWARD ボタン」と「BACK ボタン」を同時に押した場合はどのような動きをするか見てみます。このプログラムでは 2 つの「もしブロック」を必ず通ることになるので、制御カーは前進、後退を繰り返す動きをします。この様な動作を回避する、例えばどちらか一方の動作だけを実行させるにはどうすればよいか考えてみます。
この場合、もう一種類の「もしブロック」を使用します。メニューボタンから下図の「もしブロック」をワークスペースに配置してください。

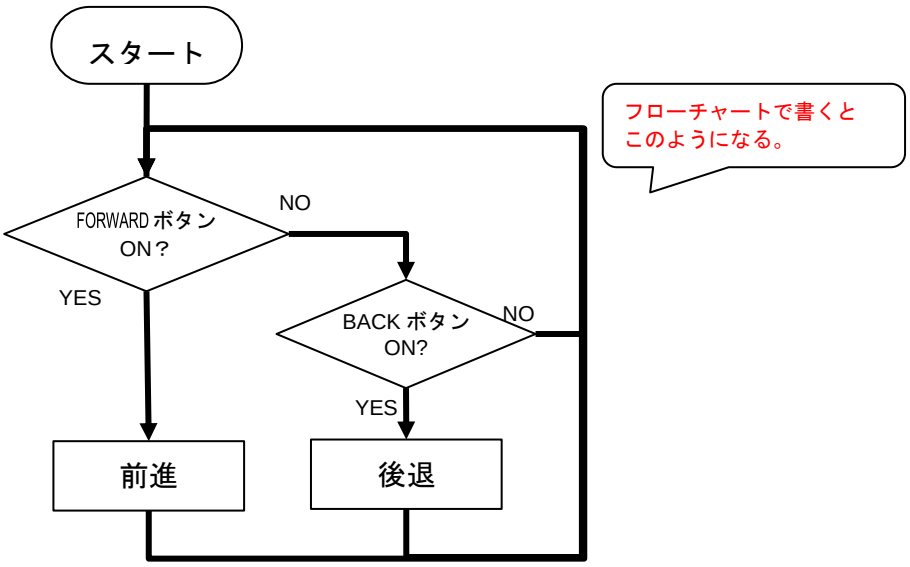


このブロックは条件が『成り立つ』場合と『成り立たない』場合でプログラムを「分岐」させるブロックです。条件が成り立つ場合は上に配置されたブロックを、成り立たない場合は下に配置されたブロックを実行します。ブロックは上か下かどちらか一方しか実行さず、他方のブロックは実行されません。





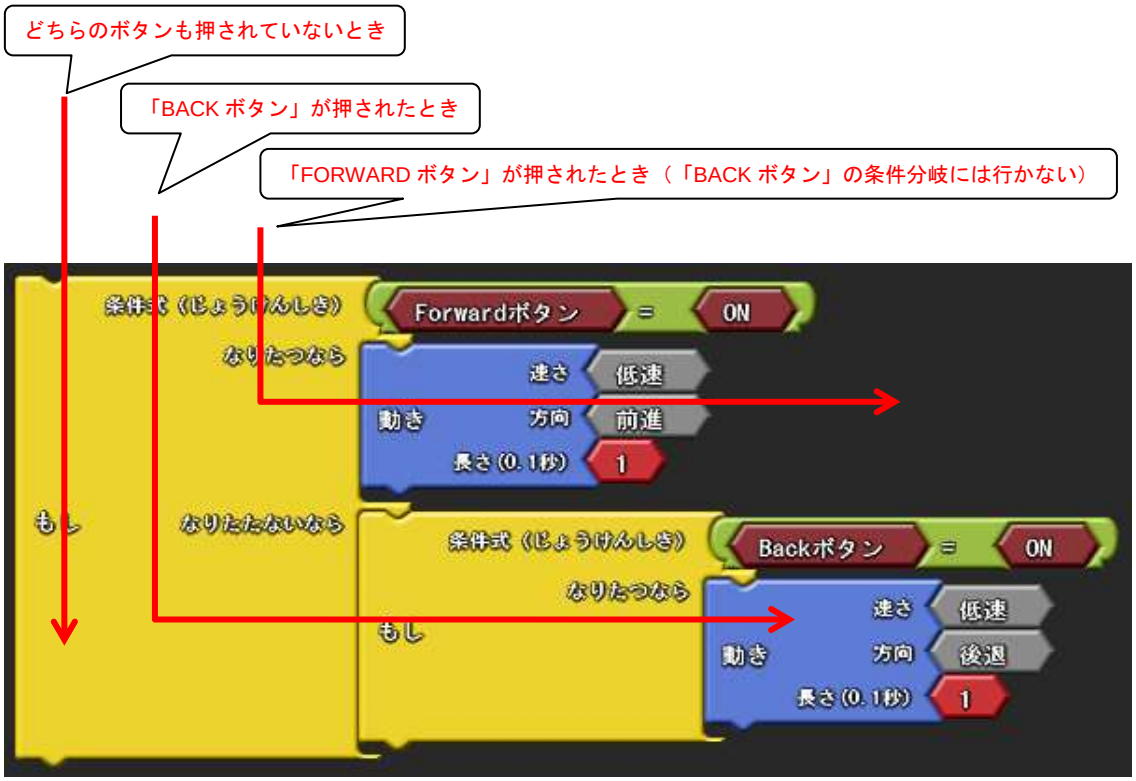
これを元に先ほどのプログラムを作り直してみます。
先ずどちらのボタンを優先させるかを決めます。今回は「FORWARD ボタン」を優先させ、同時に押した時は前進するようにします。
フローチャートで書くと下図のようになります。最初の「条件分岐」で「FORWARD ボタン」をチェックし、押されていない時だけ「BACK ボタン」のチェックに進みます。



このプログラムをブロックで作ってみます。先ず下図のようにブロックを接続し直します。「FORWARD ボタン」を条件に設定し、成り立ったとき（押されたとき）に前進するようにします。次に、成り立たない（「FORWARD ボタン」が押されていない）ときの動作を考えます。

上図のフローチャートから、成り立たないときに「BACK ボタン」の条件分岐を設定すればよいことが分かります。次ページの図のようにブロックを接続します。

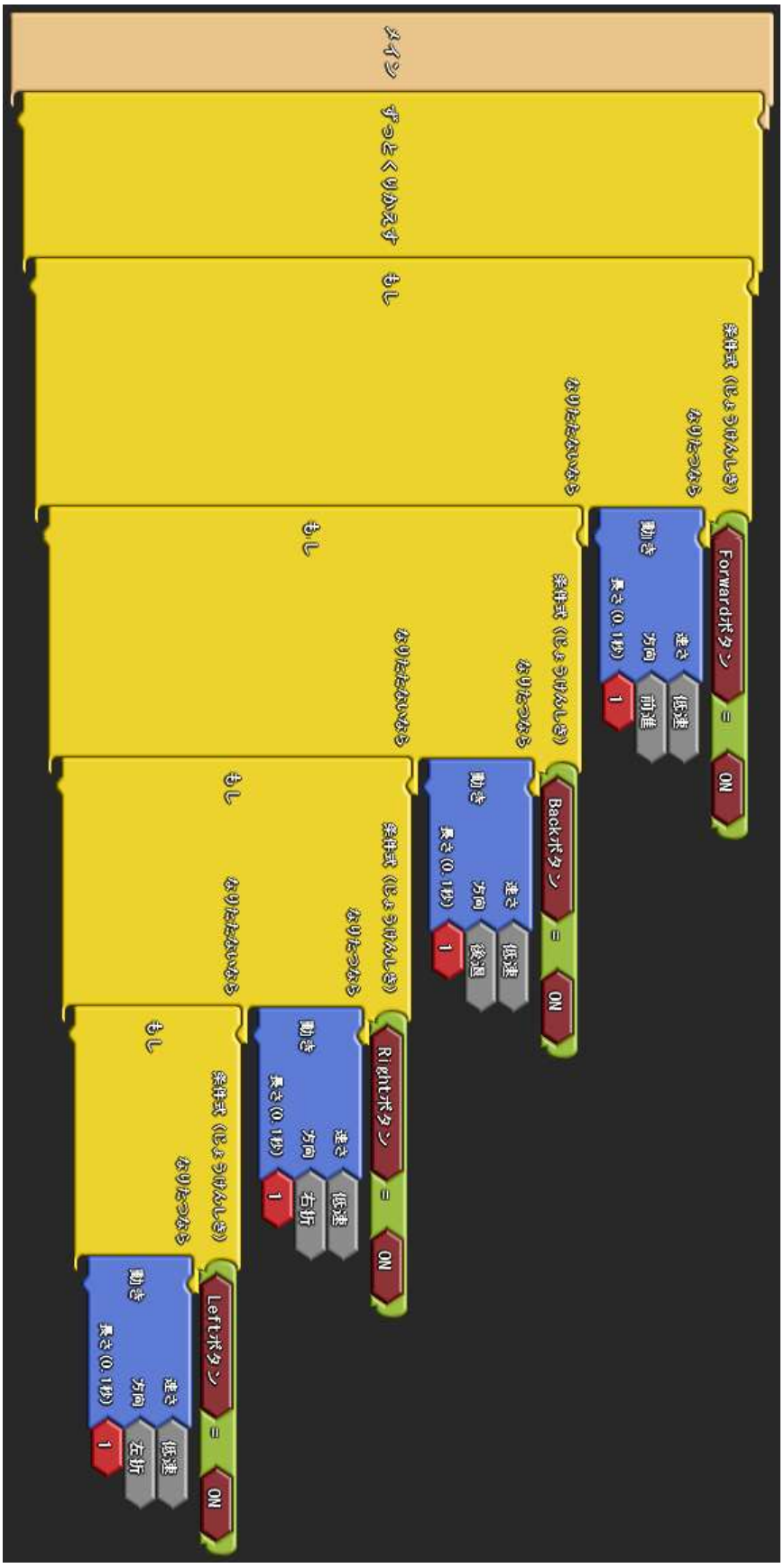
プログラムの流れは下図のようになります。



ここまでのブロック群を下図のように「くり返しブロック」と「メインブロック」に接続するとプログラムは完成です。

制御カーに書き込んで動きを確認してみましょう。

同じように「RIGHT ボタン」「LEFT ボタン」を追加したプログラムは次ページのようになります。


メイン
ずっとくりかえす
もし
条件式 (じょうけんしき)
なりたつなら
Forwardボタン
=
ON
動き
速さ
低速
方向
前進
長さ (0.1秒)
1
なりたたないなら
もし
条件式 (じょうけんしき)
なりたつなら
Backボタン
=
ON
動き
速さ
低速
方向
後退
長さ (0.1秒)
1
なりたたないなら
もし
条件式 (じょうけんしき)
なりたつなら
Rightボタン
=
ON
動き
速さ
低速
方向
右折
長さ (0.1秒)
1
なりたたないなら
もし
条件式 (じょうけんしき)
なりたつなら
Leftボタン
=
ON
動き
速さ
低速
方向
左折
長さ (0.1秒)
1

5 音を鳴らすブロック

制御カーは圧電ブザーを搭載しているので音を鳴らす事ができます。音を鳴らすには「音ブロック」を使います。

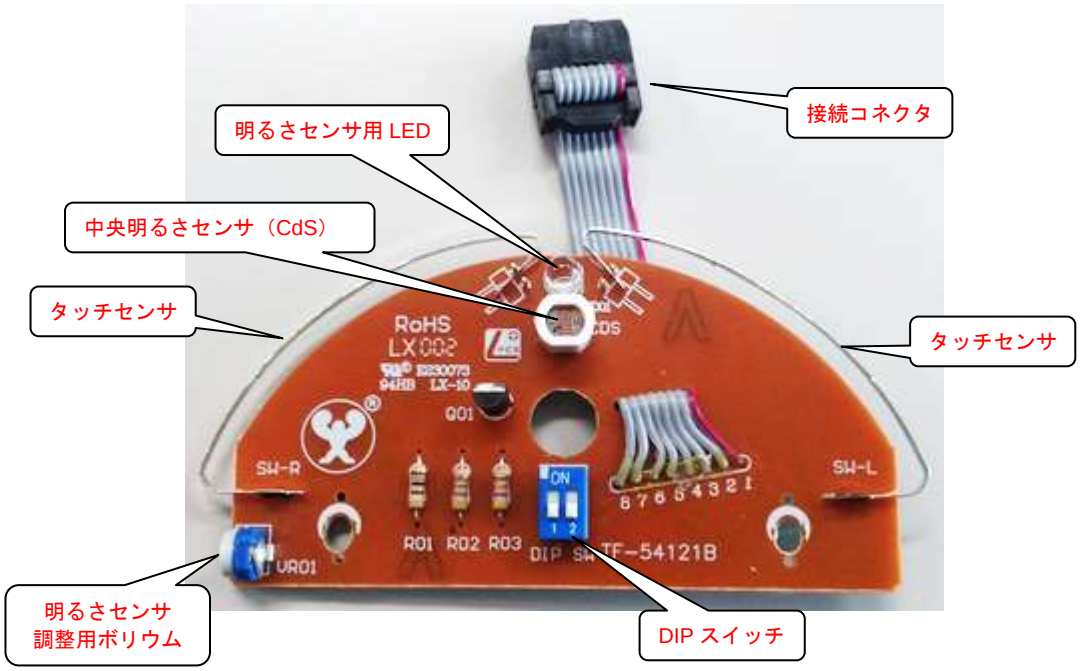
下図は「START ボタン」を押したとき音楽を鳴らすプログラム例です。

※「音ブロック」は大変大きな容量を使用するのであまり多く使用できません。

6 センサー基板を使ったプログラム

オプションのセンサー基板を使ったプログラムを作成します。

6-1 センサーオプションの説明

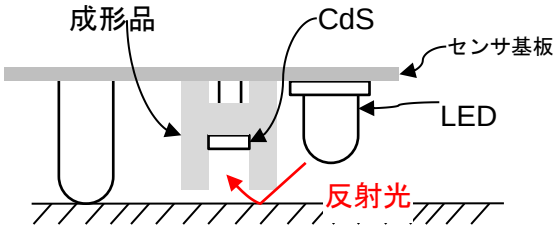


6-2 タッチセンサを使ったプログラム

ここではタッチセンサを使用したプログラムを作成します。基本的にはリモコンを使ったプログラムと同じです。条件分岐プログラムでタッチセンサを分岐条件にします。下図のプログラムでは右のタッチセンサに触れると 0.5 秒後退します。

6-3 中央明るさセンサを使ったプログラム

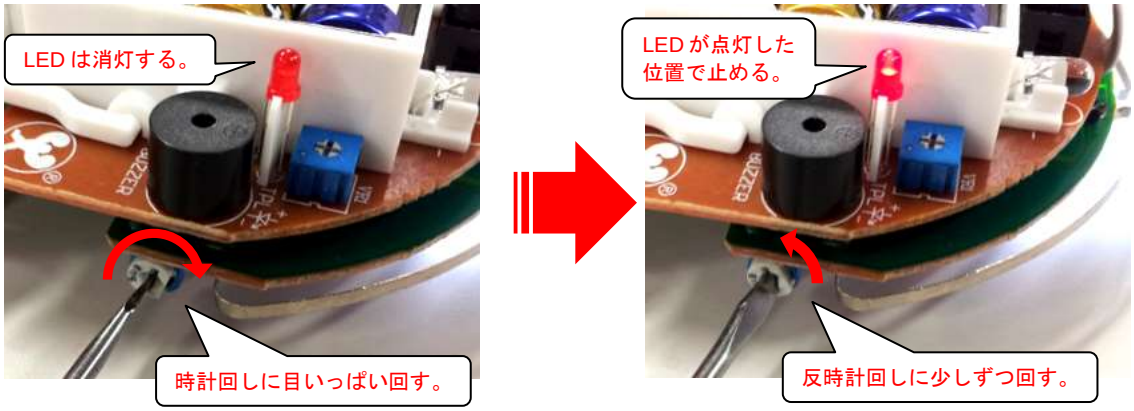
「中央明るさセンサ」は CdS を使用したセンサです。前方の LED の光の反射を捉え、地面の状態を判断します。地面が白ければ LED の反射光が CdS に届きセンサは「ON」の状態になります。逆に黒ければ反射光が届かず、センサは「OFF」となります。制御カーを持ち上げたり、机からはみ出ようとするときも反射光が届かなくなるのでセンサは「OFF」となります。



6-4 センサー感度の調整

「中央明るさセンサ」は最初に感度調整をする必要があります。先ずは「sample」フォルダ内(※)の「中央明るさセンサテスト.abp」を開き、制御カーに書き込みます。

次に 書込みライターを取り外し、制御カーを白い机（もしくは用紙、等）の上に置き、電源スイッチをいれます。そして、センサ基板の「明るさセンサ調整ボリウム」を時計回しにいっぱいまで回します。（制御カーの「チェック LED」が消灯することを確認する。）その後、ボリウムを反時計回しに少しずつ回してゆき、「チェック LED」が点灯した位置で止めます。制御カーを地面から持ち上げて「チェック LED」が消灯すれば「中央明るさセンサ」の調整は終了です。



6-5 黒い枠内から出ないプログラム

「中央明るさセンサ」を用いたプログラムは以下のようになります。ここでは説明書の裏の黒い枠から外に出ない例を示します。

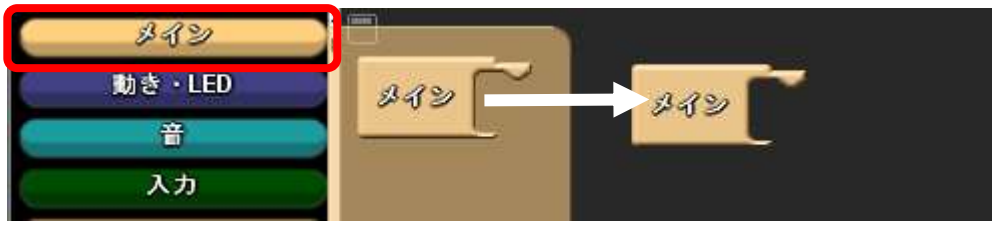
上図のプログラムでは「中央明るさセンサ」が黒い枠に差し掛かると、0.2 秒後退し 0.2 秒右折します。それ以外（床が白い）の時は、常に前進します。

7 各ブロックの説明

ここでは各ブロックの機能を説明します。

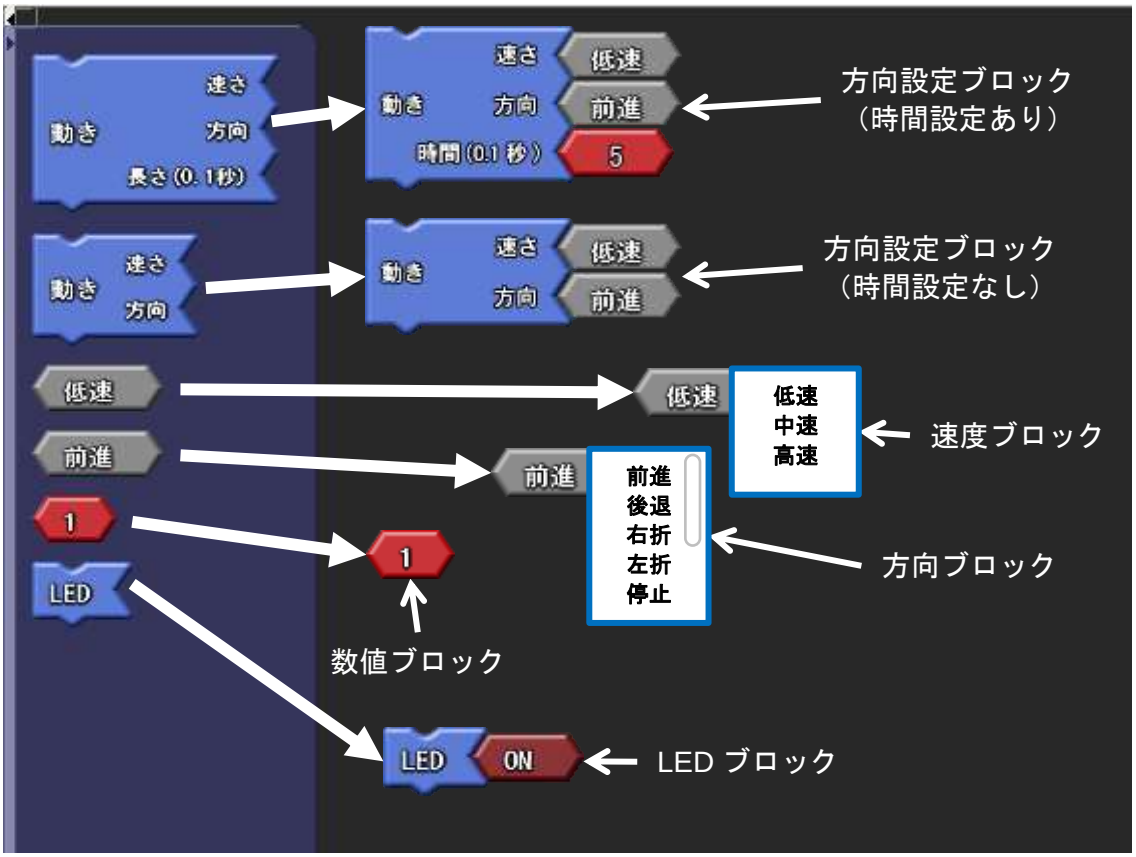
7-1 メイン

メインプログラムを作成するブロックです。このブロックの中に配置されたブロックがプログラムとして制御カーに書き込まれます。



7-2 動き・LED

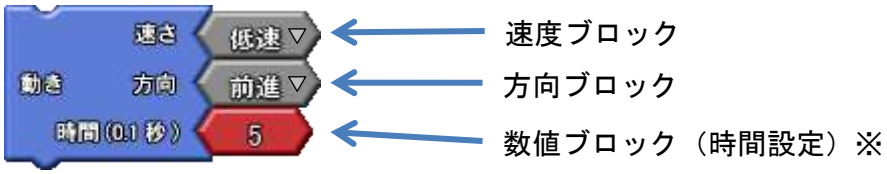
制御カーの動き及び LED の状態を設定するブロック群です。



● 方向設定ブロック（時間設定あり）

「速度ブロック」「方向ブロック」「数値ブロック（※）」を接続することで、制御カーの進む「速度」「方向」「時間」を設定します。（※設定時間経過後、制御カーは停止する。）

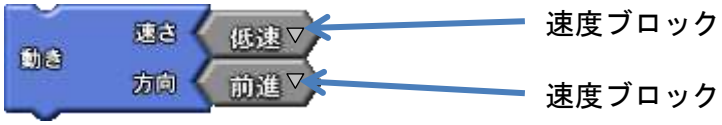
時間指定は 0.1 秒単位です。「5」とした場合は「0.5 秒」、「10」で「1 秒」となります。



● 方向設定ブロック（時間設定なし）

制御カーの進む「速度」「方向」を設定します。

時間設定が無く、メインプログラムが終了するか「停止(※次々項参照)」を指定するまで制御カーは動き続けます。

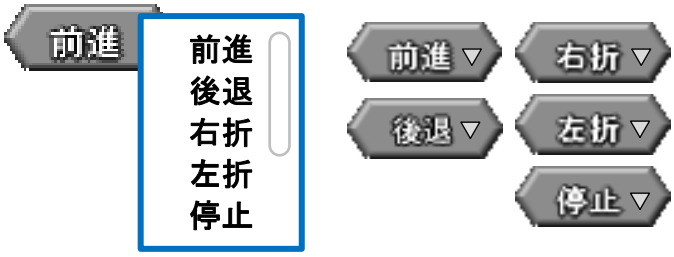


● 速度ブロック

制御カーの進む速度を「低速」「中速」「高速」の 3 段階から選択します。

● 方向ブロック

制御カーの進む方向を「前進」「後退」「右折」「左折」「停止」の５種類から選択します。

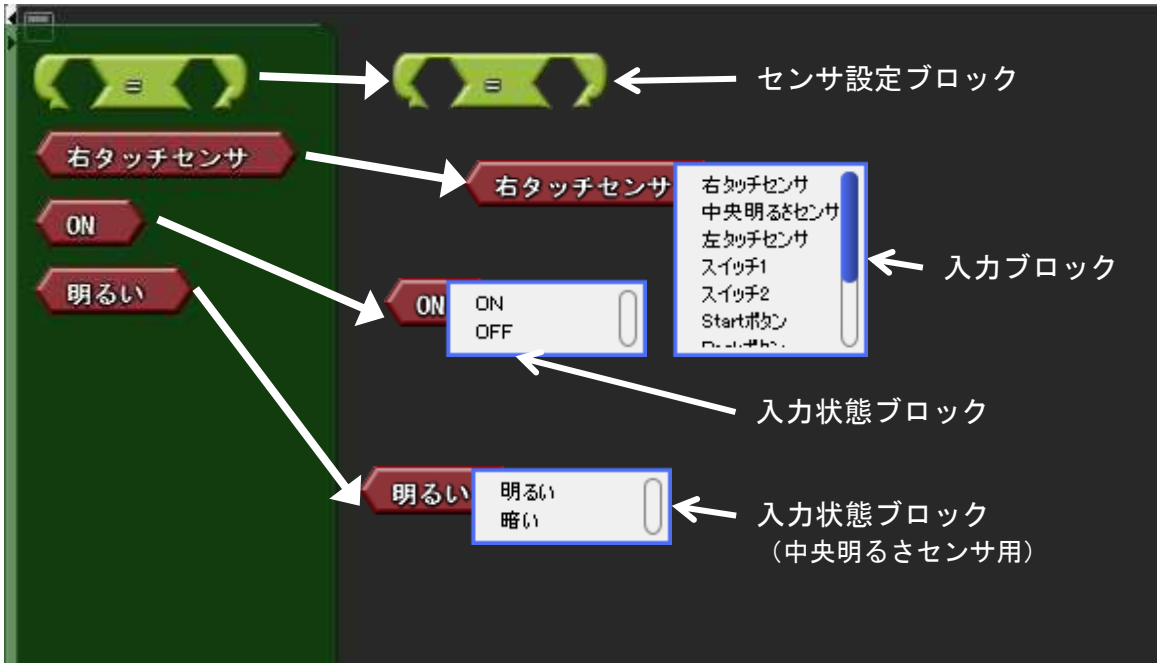


● LED ブロック

LED を点灯・消灯させます。時間を設定する場合は、「待つ」ブロックと組合わせます。

7-3 入力

各入力の状態を設定するブロック群です。

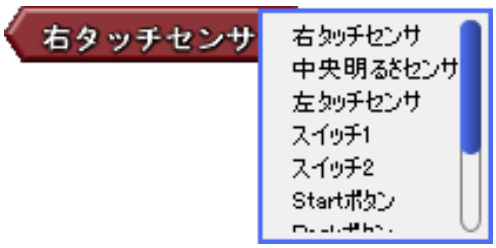


● 入力設定ブロック

「入力ブロック」「入力状態ブロック」を接続することにより、入力の状態を設定します。状態は 6 種類の中から選ぶ事ができます。

● 入力ブロック

入力は以下の 10 種類から選ぶ事が出来ます。



【センサー基板からの入力】
- **中央明るさセンサ**
- **右センサ**
- **左センサ**
- **スイッチ 1**
- **スイッチ 2**

【リモコンからの入力】
- **スタートボタン（本体と共用）**
- **Forward ボタン**
- **Back ボタン**
- **Left ボタン**
- **Right ボタン**

■センサ基板　※タッチセンサは本体底に表が下になって取りつくため、左右センサは逆になっている。

■リモコン接続状態

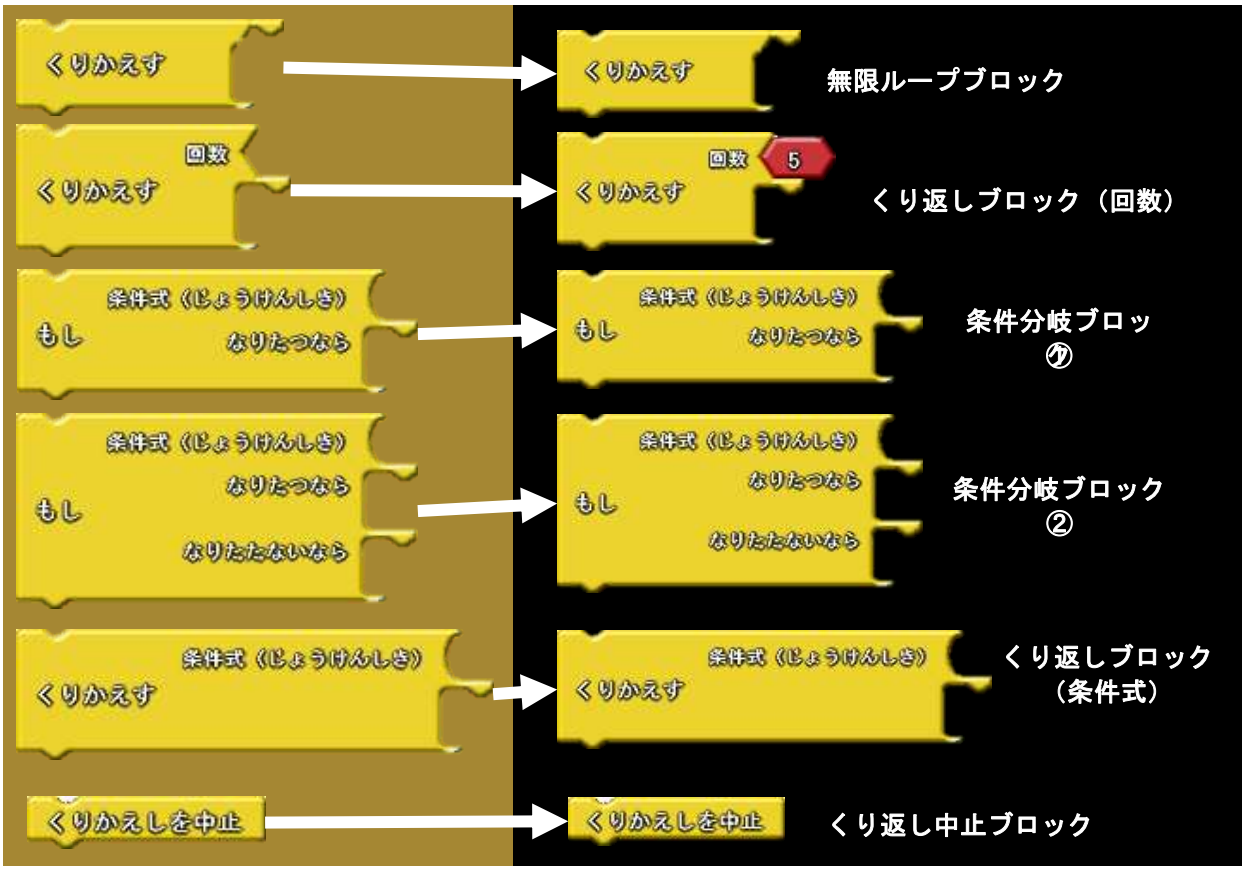
■リモコン　※スタートボタンは本体と共通。

● 入力状態ブロック

入力状態は「ON」「OFF」の２種類となります。

7-4 くりかえし・分岐

「くりかえしブロック」は２種類、「条件分岐（もし）ブロック」は２種類あります。そのほかに「条件付きのくり返しブロック」と「くり返し中止（くり返しから抜ける）ブロック」が各１種類あります。



● 無限ループブロック（ずっとくりかえす）

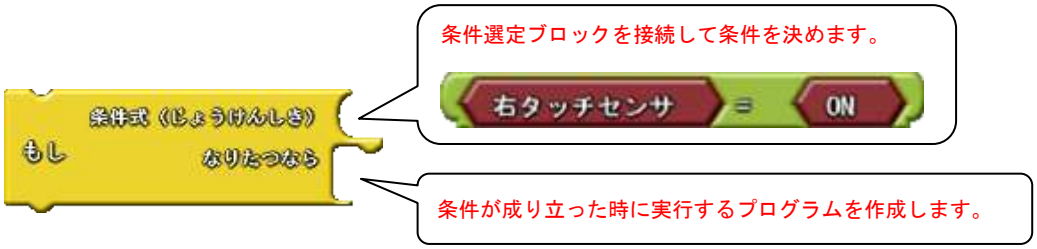
無限ループブロックは電源を切るまで接続されたプログラムを繰り返し続けます。

● くり返しブロック（回数）

「回数」を設定することでくり返し回数を決める事ができます。

● 条件分岐（もし）ブロック

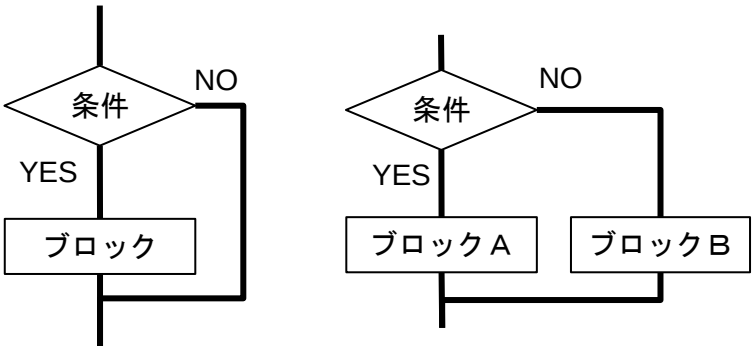
条件を設定してプログラムの流れを変える（分岐する）ブロックです。条件は入力設定ブロックを使用します。



例として「FORWARD（前進）ボタン」を押した時に制御カーが前進するプログラムを以下に示します。

条件分岐ブロックはもう一種類あります。条件に成り立つ時と成り立たない時で別々のプログラムを実行するブロックです。ブロックは上か下かどちらか一方しか実行さず、他方のブロックは実行されません。

それぞれの流れをブローチャートで書くと以下のようになります。



● くり返しブロック（条件式）

条件式を設定して、条件に合うあいだ繰り返すブロックです。

下の例は「Start ボタン」が押されるまで（OFF の状態の間は）右折を繰り返し、ボタンが押されたら停止するプログラムです。

くり返す処理を何も指定しなければ、入力があるまで次の処理に進まないプログラムが作成できます。（下の例では「Start ボタン」が押されるまで制御カーは動かない。）

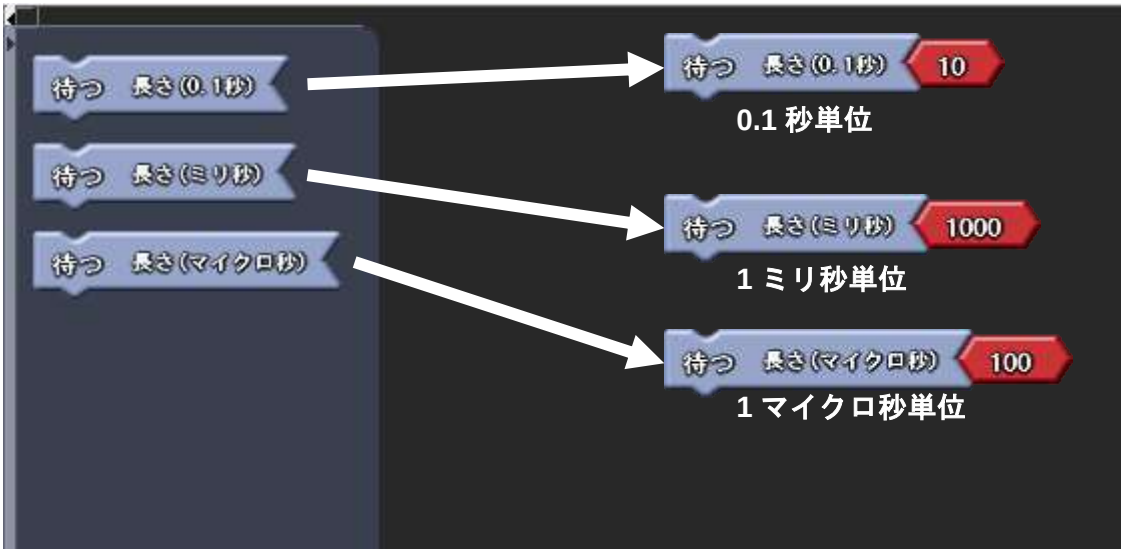
● くり返し中止ブロック

条件を設定して無限ループを抜けるブロックです。条件分岐ブロックと組合わせて使います。下の例では「Start ボタン」が押されれば、くり返しを中止し（無限ループを抜け）制御カーを動かします。

7-5 待つ

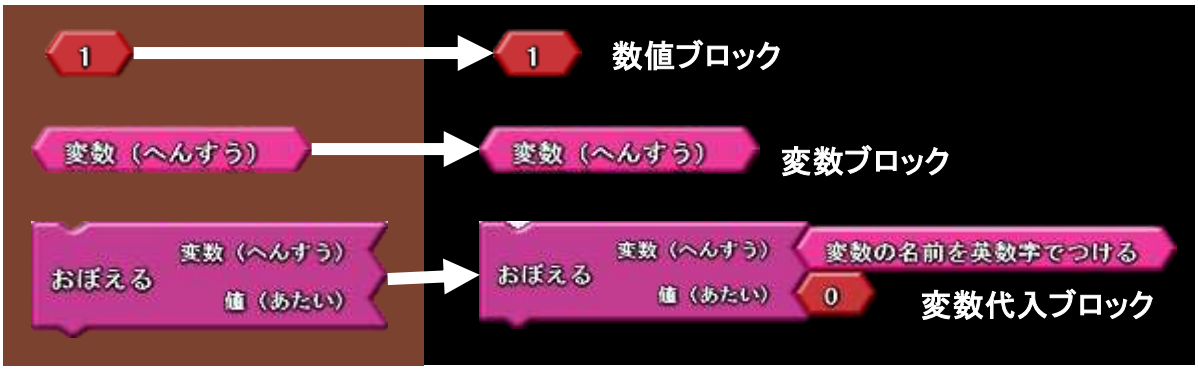
今の状態を維持したまま次の命令を待つブロックです。時間設定は「0.1 秒単位」「1 ミリ秒単位(※)」「1 マイクロ秒単位(※)」で設定ができます。

※ 1 ミリ秒=0.001 秒、1 マイクロ秒=0.000001 秒



7-6 変数

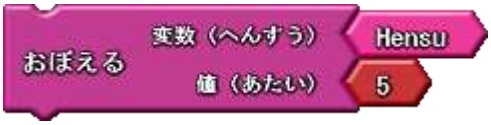
「変数」とは、プログラムで使用する数値を自由に読み書きできるためのものです。使用例として「ボタンを押した回数やセンサーに反応があった回数を記録して、〇〇回目に停止する」といったプログラムを作成する事ができます。また、同じ値を一度に変更する場合にも便利です。



変数を使用する場合、まず最初に名前を付ける必要があります。名前は 0～9、a～z、A～Z の英数字でなければなりません。また大文字と小文字は別の名前と判断されるので注意が必要です。

別の変数として判断される

変数に名前を代入するには「変数代入ブロック」を使用します。

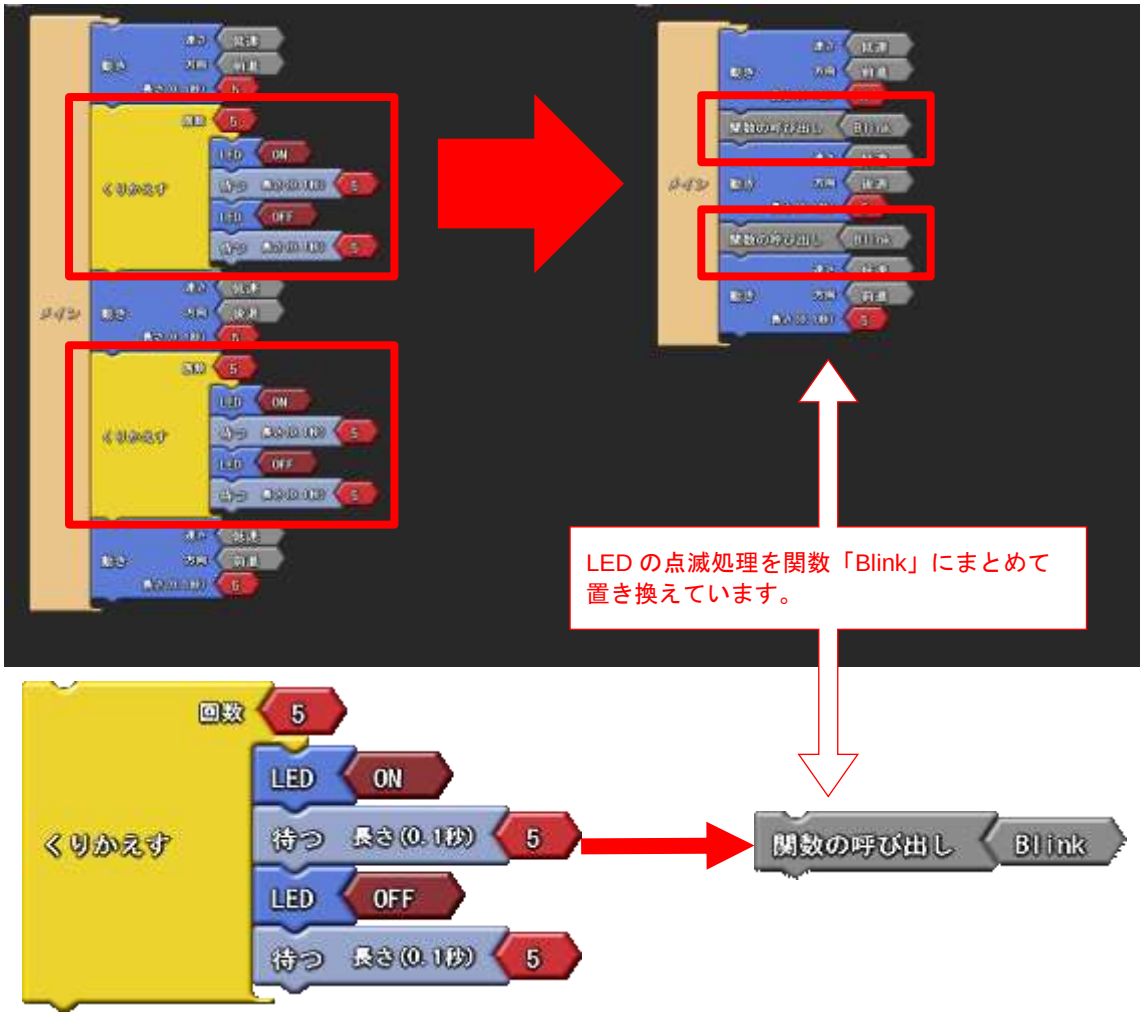


変数「Hensu」に「５」が代入される

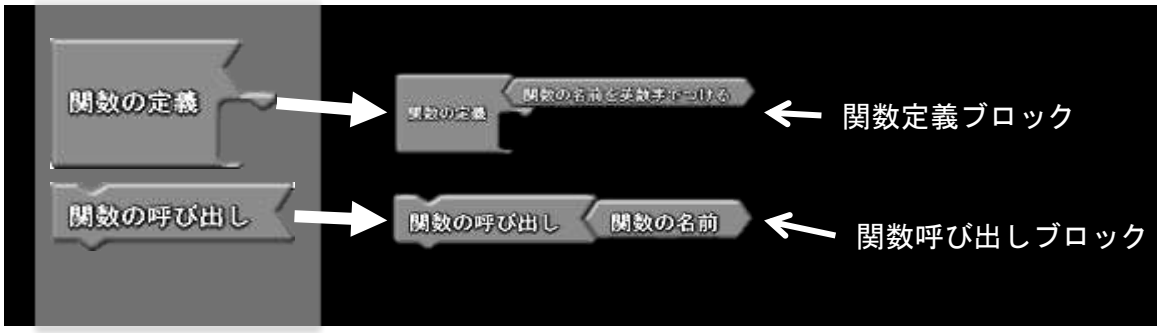
左のプログラムは変数「Hensu」に５を代入し、くり返し回数に「Hensu」を指定してます。くり返す動作は LED の点滅です。つまり、LED を「Hensu」回（5回）点滅させるプログラムになります。

7-7 関数

関数は同じ処理を一つにまとめる事ができるブロックです。
下の例では LED を 5 回点滅させる処理を一つの関数にまとめています。



関数ブロックは「呼び出しブロック」と処理の元となる「定義ブロック」の 2 種類があります。

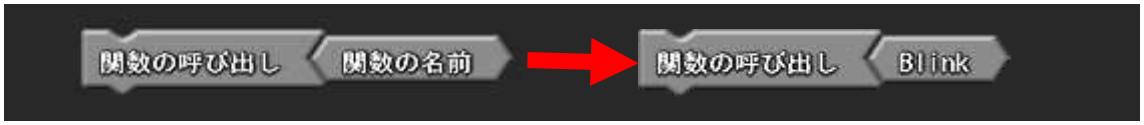


● 関数定義ブロック

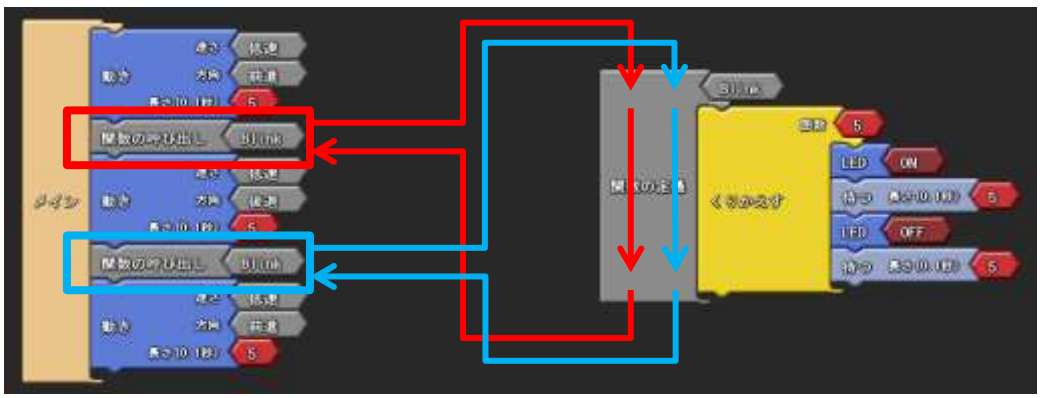
まとめる処理（プログラム）を作成するブロックです。関数には名前を付ける必要があります。名前は 0～9、a～z、A～Z の英数字でなければなりません。また大文字と小文字は別の名前と判断されるので注意が必要です。

● 関数呼び出しブロック

定義ブロックで作成した関数は「呼び出しブロック」で呼び出し使用します。呼び出しブロックの関数名は定義ブロックで付けた名前と同じにします。

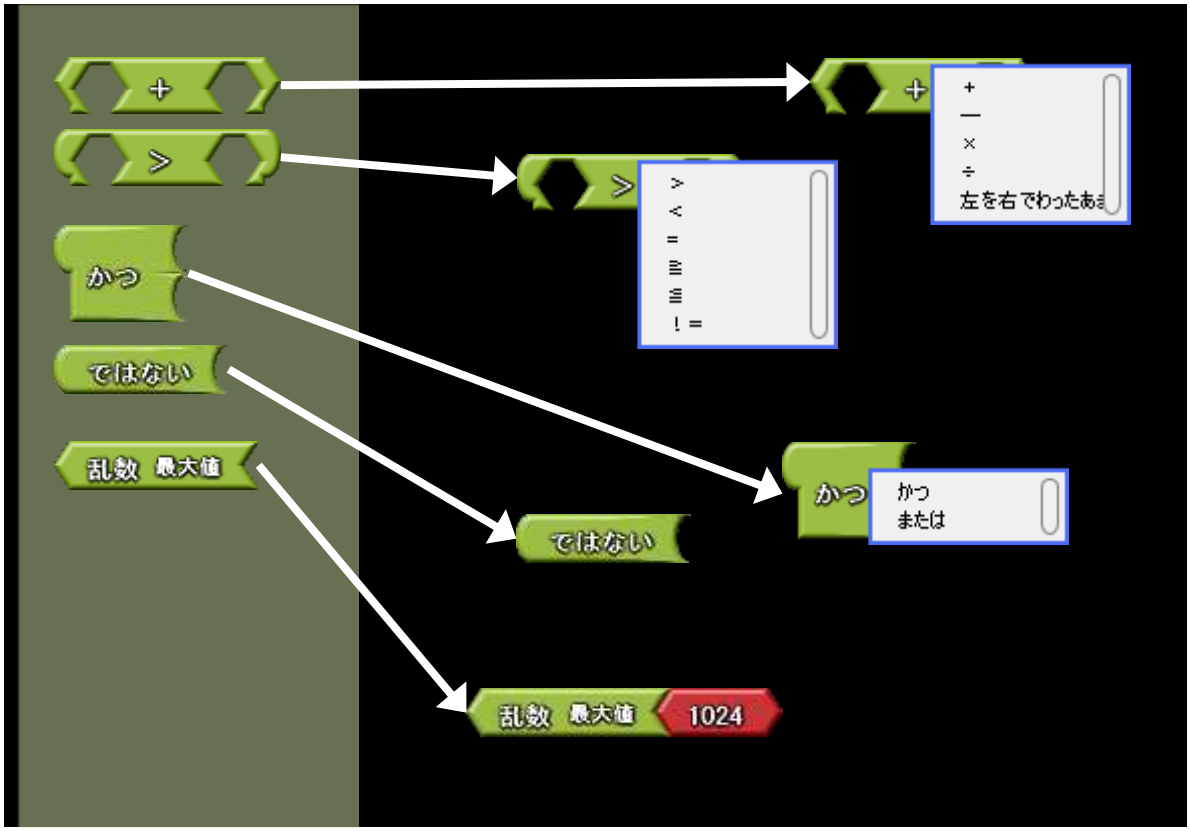


全体の構成は以下のようになります。



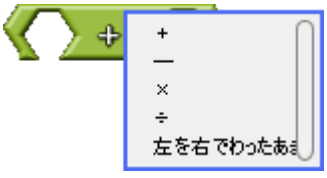
7-8 演算

演算のブロック群は以下の 5 種類で構成されています。



● 四則演算ブロック

四則演算ブロックは以下の 5 種類となります。



主に変数ブロックと組合わせて使用します。

● 等号、不等号ブロック

等号、不等号ブロックは以下の 6 種類となります。

基本的な使い方は「入力設定ブロック（P39）」と同じです。

● 集合ブロック

集合ブロックは以下の 2 種類となります。

条件分岐にて複数の条件設定を行う事ができます。「かつ」は『２つの条件がどちらも成り立つ』場合、「または」は『2 つの条件のうち、どちらかが成り立つ』場合となります。

● 補集合ブロック

補集合ブロックは条件を否定するブロックで以下の 1 種類となります。

<用法>

● 乱数ブロック

乱数ブロックはランダムな数字を設定するブロックで以下の 1 種類となります。

設定した数字の中からランダムに数字を決定します。但し、設定した数字は含まれません。「10」と設定した場合「0～9」までの 10 個の数字が対象となります。

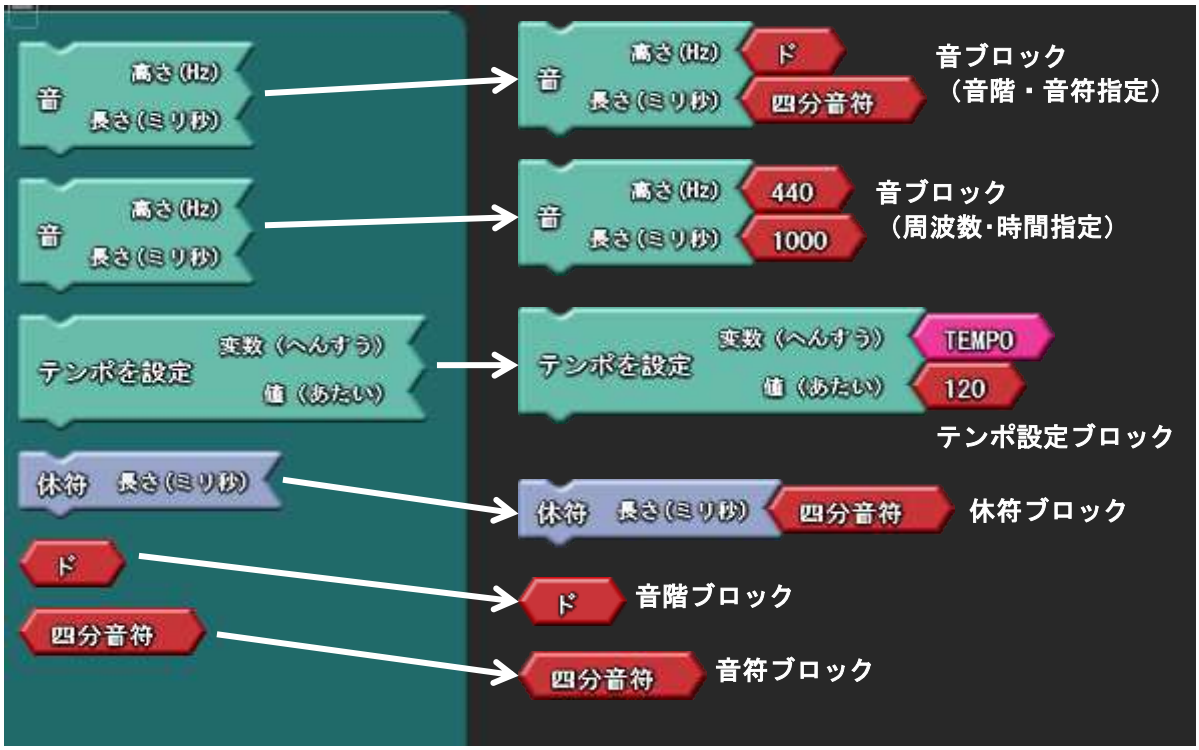
<プログラム例>

LED を点滅させる回数を乱数ブロックでランダムに設定しています。
乱数ブロックに「10」を設定しているので、乱数の範囲は「0～9」になります。
※乱数の結果が「0」の場合は LED は点滅しません。

7-9 音

制御カーの圧電ブザーから音を鳴らすためのブロックです。

※音ブロックはメモリ容量を大変多く消費するので、多く使用するとプログラムが書き込めなくなる可能性があります。



● 音ブロック

音ブロックには１音の高さと長さを音階と音符で指定するタイプと、周波数と時間で指定するタイプがあります。音の長さを音符で指定するとテンポ変更（事項にて説明）に影響されますが、時間でしてした場合は影響されません。

※音階と周波数の対応は「10-1 音階・周波数対応表」を参照ください。

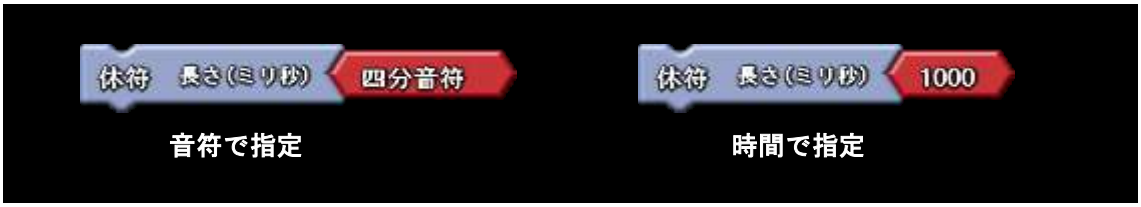
● テンポ設定ブロック

音のテンポ（音の長さ）を設定するブロックです。値を大きくするほど鳴る時間は短くなります。音ブロックで長さ音符で設定した音はこのブロックで設定したテンポに沿った長さになります。長さを時間で指定した場合、このブロックの影響は受けません。テンポはプログラムの途中で変更する事も可能です。

【注】テンポの変数名は変更しないでください。

● 休符ブロック

休符（音の出ない時間）を設定するつブロックです。
標準では音符で指定するようになっていますが、数字ブロックを使った時間指定も可能です。長さを音符で指定するとテンポ変更（事項にて説明）に影響されますが、時間でしてした場合は影響されません。



● 音階ブロック

音階は半音を含み２オクターブ（下の「ラ」から上の「ミ」まで）が指定できます。

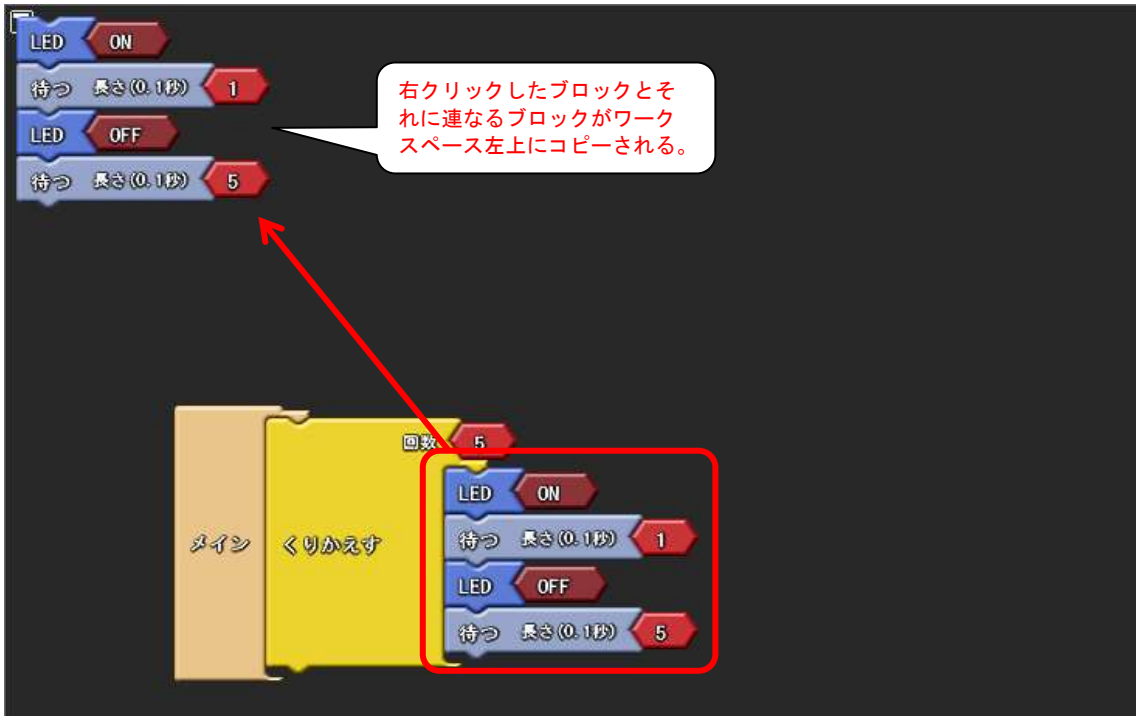
● 音符ブロック

音符は「全休符」「二分音符」「付点二分音符」「四分音符」「付点四分音符」「八分音符」「付点八分音符」「16 分音符」「付点 16 分音符」の 9 種類になります。

8 便利な機能

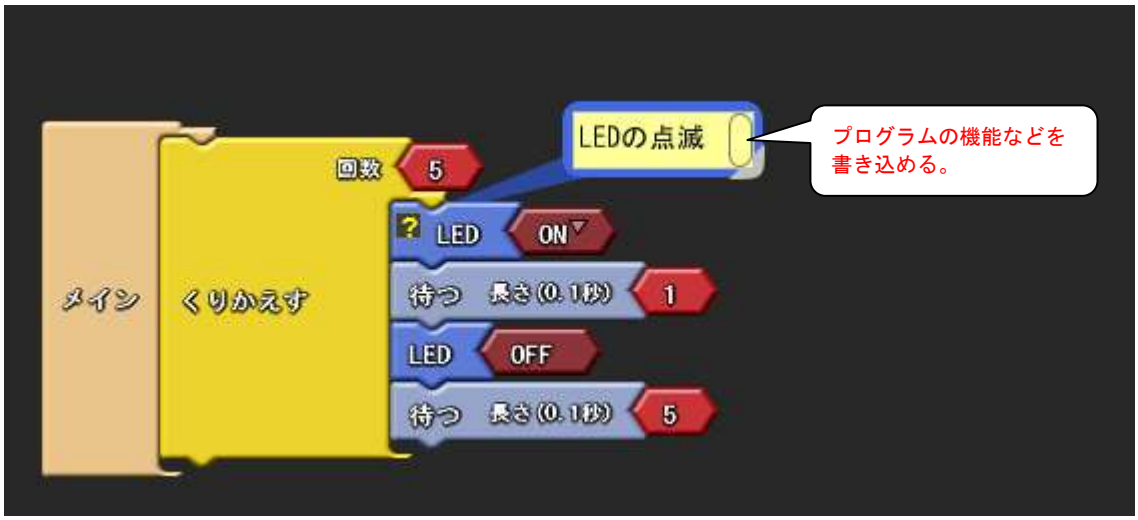
8-1 ブロックをコピーする

ブロックを右クリックして表示されるメニューより「コピー」を選択すると、右クリックしたブロックとそれに連なるブロックをコピーすることができます。



8-2 コメントを書く

ブロックにはコメントを追加する事ができます。プログラムの機能などを記入すると便利です。

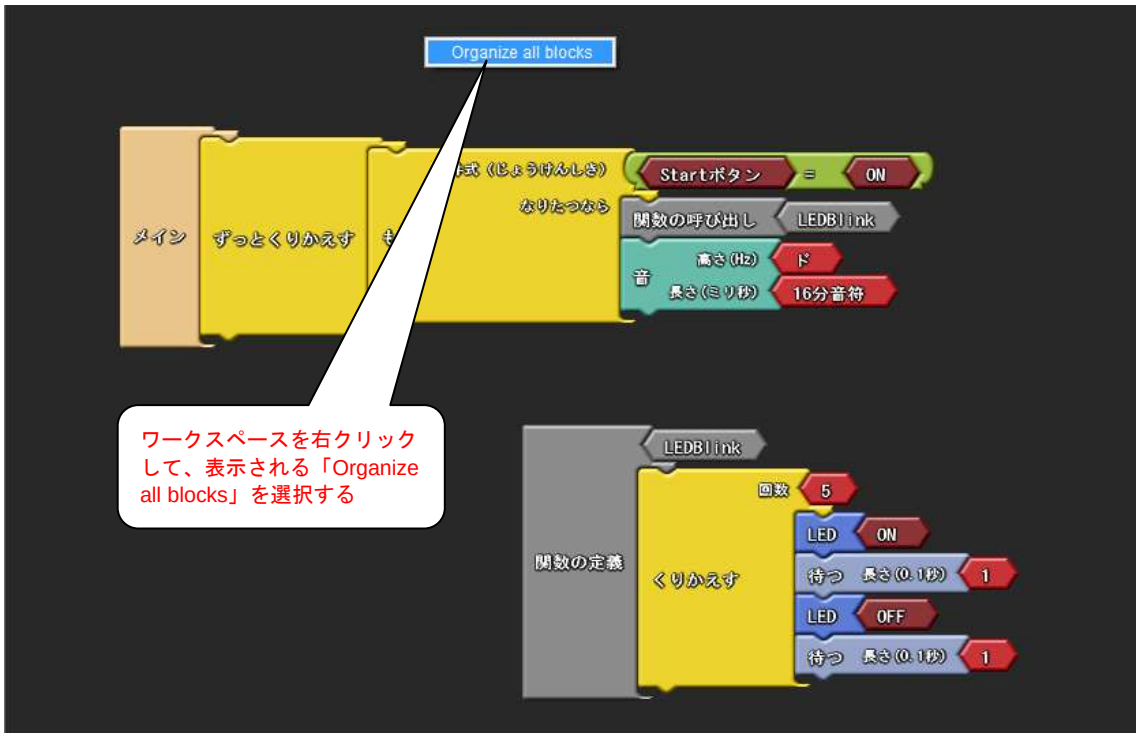


コメントを追加したブロックには?マークが表示されます。?マークをクリックするとコメントを非表示にできます。

コメントを削除するには再度ブロックを右クリックし、「Delete Comment」を選択します。

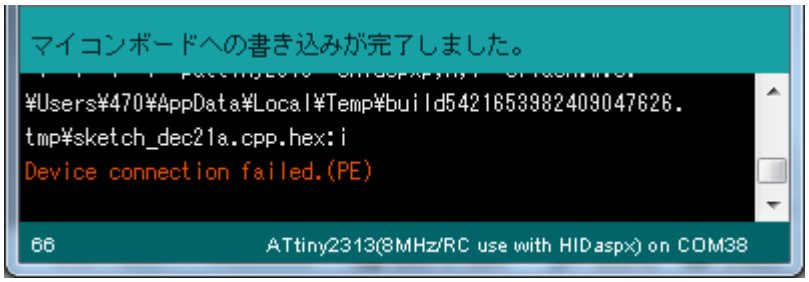
8-3 ブロックをそろえる

ワークスペースを右クリックすると表示される「Organize all blocks」を選択するとワークスペース上の全てのブロックが整列されます。

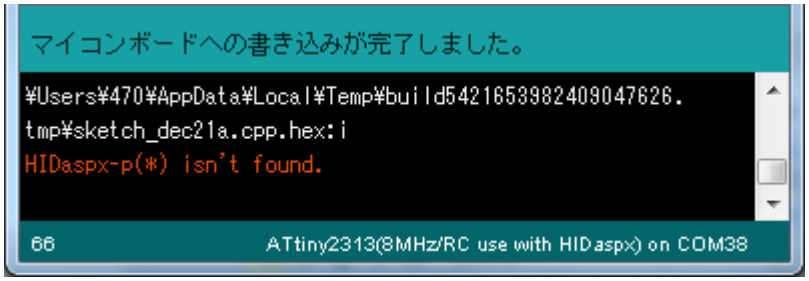


9 エラー表示

「Device connection failed.(PE)」が表示された場合は書込みライターと制御カーの接続を確認してください。



「HIDaspx-p(*) isn't found.」が表示された場合は書込みライターとパソコンの接続を確認してください。



「スケッチが大きすぎます。…」と表示された場合はプログラムがマイコンの容量を超えています。プログラムを修正して容量を減らしてください。

10　付録

10-1　音階・周波数対応表

音階	周波数(Hz)
ラ−	440
ラ#−/シ♭−	466
シ−	494

音階	周波数(Hz)
ド	523
ド#/レ♭	554
レ	587
レ#/ミ♭	622
ミ	659
ファ	698
ファ#/ソ♭	740
ソ	784
ソ#/ラ♭	831
ラ	880
ラ#/シ♭	932
シ	988

音階	周波数(Hz)
ド+	1047
ド#+/レ♭+	1109
レ+	1175
レ#+/ミ♭+	1245
ミ+	1319